KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
BADAN STANDAR, KURIKULUM, DAN ASESMEN PENDIDIKAN
PUSAT PERBUKUAN

INFORMATIKA

Auzi Asfarian, dkk

untuk SMA Kelas XI

Disclaimer: Buku ini disiapkan oleh Pemerintah dalam rangka pemenuhan kebutuhan buku pendidikan yang bermutu, murah, dan merata sesuai dengan amanat dalam UU No. 3 Tahun 2017. Buku ini digunakan secara terbatas pada Sekolah Penggerak. Buku ini disusun dan ditelaah oleh berbagai pihak di bawah koordinasi Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi. Buku ini merupakan dokumen hidup yang senantiasa diperbaiki, diperbaharui, dan dimutakhirkan sesuai dengan dinamika kebutuhan dan perubahan zaman. Masukan dari berbagai kalangan yang dialamatkan kepada penulis atau melalui alamat surel buku@kemdikbud.go.id diharapkan dapat meningkatkan kualitas buku ini.

Informatika
untuk SMA Kelas XI

Penulis
Auzi Asfarian, Paulina H. Prima Rosa, Irya Wisnubhadra, Mushtofa, Dean Apriana Ramadhan

Penelaah
Inggriani
Julio Adisantoso

Penyelia
Supriyatno
E. Oos M. Anwas
Futri F. Wijayanti

Ilustrator
Riksa Arif Fitriyani

Penata Letak (Desainer)
Tantan Yulianto

Penyunting
Lucki Hersya Rachman
Futri F. Wijayanti

Penerbit
Pusat Perbukuan
Badan Standar, Kurikulum, dan Asesmen Pendidikan
Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi
Kompleks Kemdikbudristek, Jalan RS. Fatmawati, Cipete, Jakarta Selatan
https://buku.kemdikbud.go.id

Cetakan pertama, 2021
ISBN 978-602-244-505-0 (Jil. lengkap)
ISBN 978-602-244-861-7 (jil. 2)

Isi buku ini menggunakan huruf Newsreader (Production Type/Principal design) 12/17 Pt.

xvi, 296 hlm.: 17,6 cm × 25 cm.

KATA PENGANTAR

Pusat Perbukuan; Badan Standar, Kurikulum, dan Asesmen Pendidikan; Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi sesuai tugas dan fungsinya mengembangkan kurikulum yang mengusung semangat merdeka belajar mulai dari satuan Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah. Kurikulum ini memberikan keleluasaan bagi satuan pendidikan dalam mengembangkan potensi yang dimiliki oleh peserta didik. Untuk mendukung pelaksanaan kurikulum tersebut, sesuai Undang-Undang Nomor 3 tahun 2017 tentang Sistem Perbukuan, pemerintah dalam hal ini Pusat Perbukuan memiliki tugas untuk menyiapkan Buku Teks Utama.

Buku teks ini merupakan salah satu sumber belajar utama untuk digunakan pada satuan pendidikan. Adapun acuan penyusunan buku adalah Keputusan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 958/P/2020 tentang Capaian Pembelajaran pada Pendidikan Anak Usia Dini, Pendidikan Dasar, dan Pendidikan Menengah. Sajian buku dirancang dalam bentuk berbagai aktivitas pembelajaran untuk mencapai kompetensi dalam Capaian Pembelajaran tersebut. Penggunaan buku teks ini dilakukan secara bertahap pada Sekolah Penggerak, sesuai dengan Keputusan Menteri Pendidikan dan Kebudayaan Nomor 162/M/2021 tentang Program Sekolah Penggerak.

Sebagai dokumen hidup, buku ini tentunya dapat diperbaiki dan disesuaikan dengan kebutuhan. Oleh karena itu, saran-saran dan masukan dari para guru, peserta didik, orang tua, dan masyarakat sangat dibutuhkan untuk penyempurnaan buku teks ini. Pada kesempatan ini, Pusat Perbukuan mengucapkan terima kasih kepada semua pihak yang telah terlibat dalam penyusunan buku ini mulai dari penulis, penelaah, penyunting, ilustrator, desainer, dan pihak terkait lainnya yang tidak dapat disebutkan satu per satu. Semoga buku ini dapat bermanfaat khususnya bagi peserta didik dan guru dalam meningkatkan mutu pembelajaran.

Jakarta, Desember 2021
Plt. Kepala Pusat,

Supriyatno
NIP 19680405 198812 1 001

PRAKATA

Puji Syukur pada Tuhan Yang Maha Esa karena berkat dan rahmat-Nya penulisan buku siswa mata pelajaran Informatika ini dapat diselesaikan dengan baik, dan semoga dapat dimanfaatkan oleh siswa Kelas XI untuk belajar bermakna tentang informatika, sebagai salah satu mata pelajaran pilihan yang dapat dipilih oleh siswa yang akan melanjutkan studi di perguruan tinggi rumpun Informatika seperti yang dijelaskan dalam Buku Informatika Kelas X, atau studi lanjut yang akan memerlukan banyak komputasi atau *engineering*.

Tidak dapat dipungkiri bahwa di era industri 4.0 dan Masyarakat 5.0 saat ini, semua bidang akan bersentuhan dengan komputer. Pandemi yang melanda dunia pada tahun 2020 membuat kita semakin menghadapi dunia VUCA (*Volatility*, *Uncertainty*, *Complexity*, *Ambiguity*). Pengalaman menghadapi pandemi bersama seluruh warga dunia telah menyadarkan kita bahwa manusia semakin tergantung kepada komunikasi secara daring, yang membutuhkan kecakapan literasi digital dalam berkomunikasi dan berkolaborasi untuk memecahkan persoalan yang semakin kompleks, di sebuah dunia digital yang global, aman dan nyaman. Selain keterampilan memecahkan persoalan dan teknis, isu mengenai *privacy*, *security* semakin membutuhkan pemikiran kritis. Namun, literasi digital apalagi hanya TIK saja tidak cukup. Dibutuhkan lebih banyak pencipta produk-produk komputasional yang sangat dibutuhkan di masa mendatang. Oleh sebab itu, dengan bonus demografi yang terjadi, peserta didik Indonesia melalui pengetahuan dan keterampilan Informatika, diharapkan lebih banyak berkontribusi kepada dunia, termasuk karya komputasional di bidang studi apapun yang akan ditempuh.

Selain semakin tergantung kepada aplikasi dan sistem komputasi, manusia abad kini semakin menyadari permasalahan yang terjadi akibat penerapan teknologi bukan hanya teknologi informasi dan komunikasi saja. Warga dunia semakin peduli terhadap keamanan dirinya dan orang lain saat menggunakan teknologi.

Akhir kata, penulis berharap semoga buku siswa ini dapat memberikan manfaat dan dapat digunakan untuk pendamping belajar Informatika sebaik-baiknya. Saran dan kritik membangun sangat penulis harapkan untuk perbaikan penulisan buku lebih lanjut.

Jakarta, Desember 2021

Tim Penulis.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

PETUNJUK PENGGUNAAN BUKU SISWA KELAS XI

Buku teks pelajaran Informatika dirancang sebagai pendamping peserta didik dalam melakukan pembelajaran. Buku ini berisi konsep dan aktivitas yang merangsang peserta didik untuk dapat belajar dan melakukan eksplorasi secara mandiri. Bagian-bagian penting dari buku ini dapat dijadikan acuan bagi peserta didik untuk belajar adalah:

Tujuan pembelajaran berisi kompetensi utama yang akan dicapai pada bab tertentu pada buku. Kalian diharapkan mencapai kompetensi yang telah ditetapkan tersebut setelah proses pembelajaran pada bab tersebut selesai.

Pertanyaan pemantik berisi pertanyaan yang memantik kalian untuk berpikir dan menjawab pertanyaan mengenai hal baru yang akan dipelajari. Kalian dapat mendiskusikan jawaban tersebut dengan guru.

Peta Konsep berisi gambaran umum atau abstraksi dari konsep yang akan kalian pelajari pada tiap bab. Peta konsep berbentuk hirarki yang menunjukkan hubungan antar konsep.

Apersepsi merupakan cerita awal atau kegiatan yang diberikan kepada kalian untuk mendekatkan konsep dan pengetahuan baru agar kalian dapat memusatkan perhatian pada sepenuhnya pada pengetahuan baru tersebut.

Kata Kunci adalah daftar kata penting yang akan kalian pelajari pada tiap bab.

Aktivitas adalah kegiatan pembelajaran berbasis peserta didik, dimana kalian akan melakukan kegiatan untuk membentuk kompetensi yang telah ditetapkan pada tiap bab. Aktivitas ini dapat dilakukan secara individu, berpasangan, maupun kelompok untuk mendapatkan keseimbangan kompetensi dalam hal kemandirian dan gotong-royong yang sering dilakukan pada kegiatan pengembangan produk informatika. Guru akan memberikan penjelasan mengenai aktivitas ini dan kalian diharapkan dapat bereksplorasi secara maksimal.

Buku Kerja adalah buku catatan peserta didik yang digunakan untuk mendokumentasikan seluruh hasil pekerjaan saat melaksanakan aktivitas di Informatika Kelas XI.

Lembar Refleksi adalah salah satu bagian di Buku Kerja yang digunakan peserta didik sebagai catatan pribadi atas aktivitas refleksi di akhir pembelajaran. Cek Buku Informatika Kelas X untuk melihat contoh lembar refleksi.

Jurnal Peserta Didik adalah buku catatan pengalaman belajar peserta didik. Cek Buku Informatika Kelas X untuk melihat contoh jurnal peserta didik (Bab 1).

Asesmen Diagnostik adalah salah satu cara bagi guru dan peserta didik untuk menilai kesiapan mereka dalam mengikuti Informatika Kelas XI. Asesmen Diagnostik disarankan untuk diberikan apabila peserta didik yang akan mengambil Informatika Kelas XI belum mengambil Informatika di jenjang kelas sebelumnya.

Ayo Membaca! adalah aktivitas yang memandu peserta didik untuk membaca suatu bahan bacaan yang diberikan dan memetik hikmah dan pelajaran dari bacaan tersebut.

Ayo Berdiskusi! adalah aktivitas yang memandu peserta didik untuk saling berdiskusi pada suatu topik yang diberikan. Di akhir diskusi, kalian akan membagi hasil diskusi kalian dengan kelompok diskusi lainnya.

Ayo Berlatih! adalah salah satu jenis aktivitas pembelajaran untuk melatih peserta didik menguasai tujuan pembelajaran yang diberikan. Kalian diharapkan berlatih untuk menyelesaikan masalah yang ada dengan cara yang efektif dan efisien.

Ayo Merancang Program! Adalah aktivitas yang memandu peserta didik secara berpasangan untuk merancang suatu program sebagai suatu solusi dari permasalahan atau problem yang diberikan. Hasil rancangan berupa algoritma yang telah dicek dan siap diimplementasikan dalam bentuk program.

Ayo Buat Program! Adalah aktivitas yang memandu peserta didik secara berpasangan untuk membuat suatu program berdasarkan algoritma dan mengujinya.

Ayo Kerjakan! Adalah aktivitas yang memandu peserta didik untuk mengerjakan suatu instruksi yang diberikan guna mencapai pengalaman belajar dan capaian pembelajaran yang diharapkan.

Ayo Bertanya! Adalah aktivitas yang memandu peserta didik untuk merumuskan pertanyaan kritis atas suatu pokok bahasan yang sebelumnya dibaca dari suatu artikel.

Ayo Lakukan! Adalah aktivitas yang memandu peserta didik untuk mengerjakan suatu kegiatan terstruktur

Ayo Kembangkan! Adalah aktivitas untuk mengembangkan aplikasi dari aplikasi yang sudah ada. Pengembangan aplikasi untuk menambah fitur atau fungsionalitas dari aplikasi.

Ayo Berpikir! Adalah aktivitas menggunakan akal budi untuk membentuk konsep, memecahkan masalah, melakukan penalaran, atau membuat keputusan.

Ayo Renungkan! adalah aktivitas refleksi yang dikerjakan di akhir aktivitas, subbab, atau bab pada Informatika Kelas XI. Refleksi ini berfungsi untuk merekatkan pengalaman belajar kalian dalam ingatan jangka panjang yang akan bermanfaat bagi kalian suatu hari nanti.

Aktivitas PLB! adalah aktivitas yang pelaksanaannya akan melibatkan bidang lain, tidak hanya informatika.

Mari Sadari! adalah sebuah tulisan singkat di aktivitas pada Bab Berpikir Kritis dan Dampak Sosial Informatika yang memberikan petunjuk bagi peserta didik untuk memahami lebih dalam pengalaman pembelajaran yang diberikan pada aktivitas.

Refleksi adalah kegiatan untuk merenungkan apa yang kalian pelajari, bagaimana proses pembelajaran terjadi, dan kalian dapat memberikan umpan balik atas apa yang telah dilakukan untuk memperbaiki efektivitas proses pembelajaran.

Tahukah Kalian?

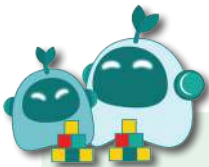
Tahukah Kalian! adalah bagian dalam buku yang ditulis untuk menginformasikan tentang fakta-fakta menarik dan penting tentang topik yang dipelajari.

Ini Berpikir Komputasional!

adalah bagian dalam buku yang diberikan untuk menekankan aspek Berpikir Komputasional dalam topik yang dibahas.

Ini Informatika

adalah sebuah tulisan singkat di akhir aktivitas pada Bab Strategi Algoritmik dan Pemrograman yang akan menghubungkan pengalaman belajar dengan elemen-elemen Informatika.

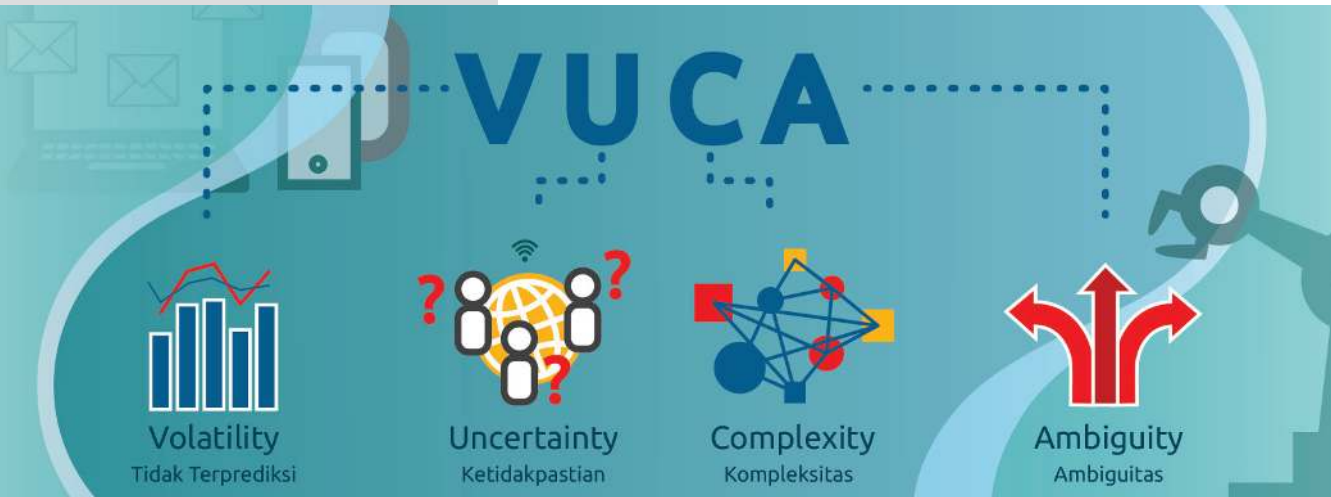
Bagian **Uji Kompetensi** dan **Asesmen** ini berisi soal-soal untuk menguji kompetensi kalian sesuai dengan bab pelajaran.

Pengayaan

berisi konsep, bahan, dan kegiatan yang dapat kalian lakukan jika kalian tertarik untuk mempelajari lebih lanjut tentang bab yang telah selesai. Kalian dapat berdiskusi dengan guru kalian tentang hal ini.

Ingin Tahu Lebih?

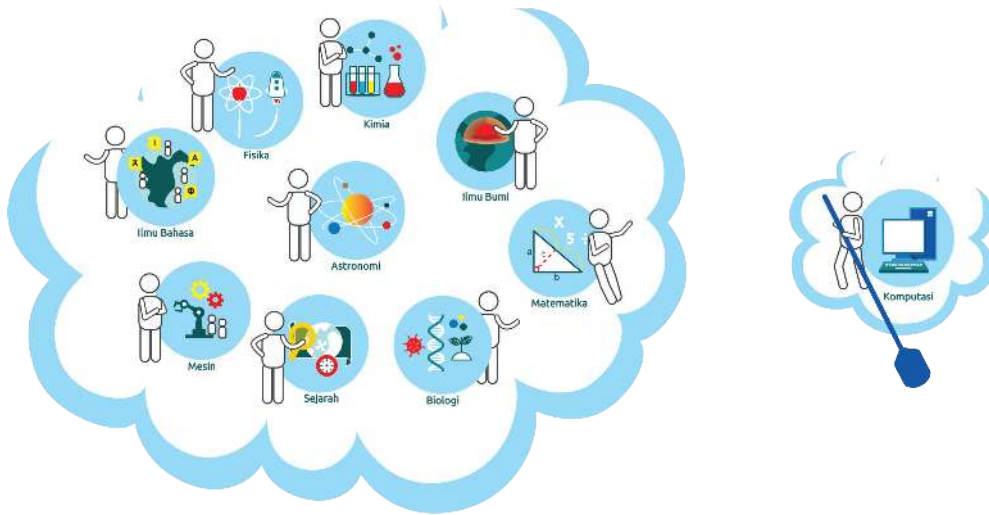
Kegiatan yang dapat kalian lakukan jika kalian tertarik untuk mempelajari lebih lanjut tentang bab yang telah selesai. Kalian dapat berdiskusi dengan guru kalian tentang hal ini.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas XI
Penulis : Auzi Asfarian
Paulina H. Prima Rosa
ISBN: 978-602-244-861-7

1 Tentang Informatika

Tujuan Pembelajaran

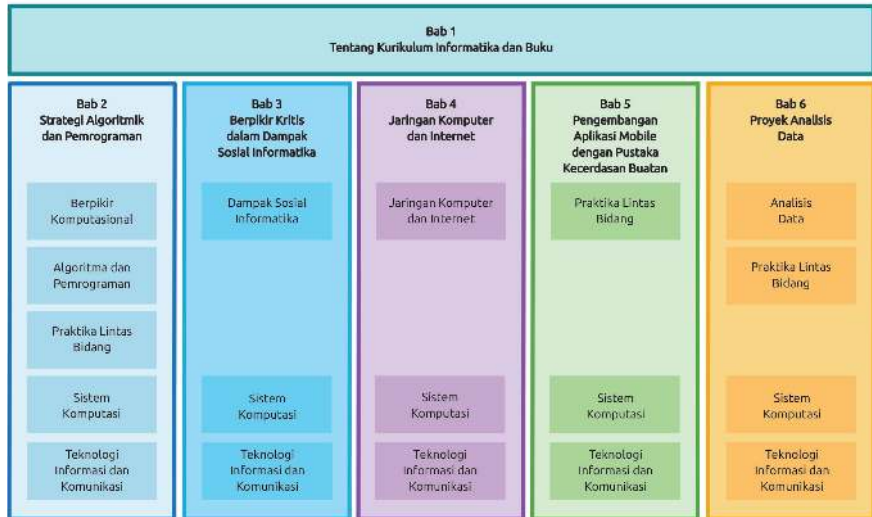
Jika kalian mengikuti tahapan pembelajaran informatika yang sesuai dengan urutan jenjangnya, tentu kalian sudah menerima pembelajaran Informatika dari kelas VII, VIII, IX, dan X sebelum mulai memakai buku ini. Namun, jangan khawatir jika kalian langsung mendapatkan materi informatika di Kelas XI, karena dalam pembelajarannya materi tersebut akan disesuaikan oleh guru kalian dengan menggabungkannya dengan materi Informatika di Kelas X yang dibutuhkan untuk merasakan pengalaman belajar Informatika di Kelas XI. Jadi, tetap semangat, ya!

Setelah mempelajari bab ini, kalian diharapkan mampu memahami gambaran umum mengenai hal apa saja yang akan dipelajari pada mata pelajaran Informatika Kelas XI sebagai mata pelajaran pilihan.

.

Gambar 1.1 *The Great Principles of Computing*

Pertanyaan Pemantik

Mengapa kalian memilih Informatika sebagai mata pelajaran pilihan di Kelas XI ini? Apakah kalian benar-benar memahami konsekuensi (materi pelajaran, kelanjutan studi, karir, dan sebagainya) yang akan dihadapi ke depannya jika memilih mata pelajaran ini?

Peta Konsep

Gambar 1.2 Peta Konsep Tentang Kurikulum Informatika dan Buku Apersepsi Aktivitas Individu

Ayo, Kita Membaca!

Aktivitas Individu

Aktivitas IF-K11-01-U: Bacaan Singkat Informatika

Melalui bacaan singkat ini, kalian diminta untuk memahami secara umum tentang Mata Pelajaran Informatika untuk Kelas XI. Pemahaman umum ini penting untuk menjadi peta perjalanan kalian menempuh mata pelajaran ini sepanjang tahun. Ingatlah bahwa Mata Pelajaran Informatika di Kelas XI adalah mata pelajaran pilihan. Jika mata pelajaran wajib dirancang karena memang diperlukan untuk semua, maka mata pelajaran pilihan dirancang untuk kalian yang benar-benar berminat untuk mendalami bidang yang dipilih. Kalian perlu menyadari pentingnya Informatika untuk mendukung kelanjutan studi dan masa depan kalian. Setiap pilihan akan mengandung konsekuensi dan karena merupakan pilihan kalian sendiri setelah berdiskusi dengan guru dan orang tua/wali, hendaknya dijalani dengan bersungguh-sungguh dan penuh motivasi.

Tentang Informatika

Di era industri 4.0 dan masyarakat 5.0, Informatika merupakan salah satu disiplin ilmu yang wajib dikuasai oleh setiap orang, dan aspek praktisnya sudah diperlukan sejak usia dini. Belajar Informatika tidak berarti harus selamanya di depan komputer. Informatika adalah sebuah disiplin ilmu yang mencari pemahaman dan mengeksplorasi dunia di sekitar kita, baik natural (dunia dan alam sekitar kita) maupun artifisial (dunia maya atau dunia digital yang diciptakan manusia). Informatika juga berkaitan dengan studi, pengembangan, dan implementasi dari sistem komputer, serta pemahaman terhadap prinsip-prinsip dasar pengembangan yang didasari pada pemahaman dunia nyata dan dunia artifisial tersebut. Ilmu informatika tidak eksklusif, tetapi banyak bersinggungan dengan bidang ilmu lain karena luasnya kemungkinan eksplorasi masalah yang akan dipecahkan.

Kurikulum Informatika terdiri atas 8 elemen, yaitu Berpikir Komputasional (BK), Teknologi Informasi dan Komunikasi (TIK), Sistem Komputer (SK), Jaringan Komputer dan Internet (JKI), Analisis Data (AD), Algoritma dan Pemrograman (AP), Dampak Sosial Informatika (DSI), dan Praktik Lintas Bidang (PLB), seperti digambarkan dalam "rumah" Informatika sebagai berikut.

Gambar 1.3 Pilar Pengetahuan Informatika

Kedelapan elemen tersebut erat kaitannya satu sama lain. Semua elemen pengetahuan tersebut akan dipelajari melalui aktivitas yang dirancang pada buku ini. Harapannya, peserta didik dapat memperdalam dan memperluas konsep serta menambah keterampilan yang telah dijalani sesuai dengan materi dalam buku ini. Setelah mempelajari dan menjalani aktivitas pada mata pelajaran Informatika sejak Kelas VII sampai dengan Kelas X yang diorganisasikan dengan modular per elemen, materi Informatika Kelas XI akan lebih banyak dipraktekkan secara terintegrasi. Mengapa demikian?

Selain berakar ke ilmu formal, Informatika juga berakar ke *engineering*. Informatika hakekatnya mengemas STE(A)M: *Science* (Sains), *Technology* (Teknologi), *Engineering* (Teknik), *Art* (Seni) and *Math* (Matematika). STEAM itu erat kaitannya satu sama lain. Membuat sebuah aplikasi adalah menciptakan artefak komputasional, baik itu hanya perangkat lunak, atau merupakan gabungan perangkat keras dan perangkat lunak (sebuah sistem komputer). Membangun sebuah artefak komputasional untuk bidang tertentu misalnya pertanian juga perlu memperhatikan sains yang terkait pertanian (biologi, ilmu tanah atau media tanamannya) selain sains informatika. Teknologi (perangkat TIK dan perangkat lainnya) dan engineering (proses menganalisis dan merancang) dibutuhkan baik untuk mengembangkan produk, maupun untuk memakai produk. Art atau aspek seni dibutuhkan bukan hanya untuk produk digital kreatif, tetapi bagaimana merancang produk yang nyaman dipakai serta indah dipandang. Matematika konon disebut: "*Queen of Science*" yang mendasari semua model komputasi dari produk komputasional apapun.

Lalu apa hubungannya dengan Informatika Kelas XI? Pembelajaran Informatika Kelas XI tidak lagi diorganisasi per elemen, tetapi dikemas sesuai dengan elemen yang dicakup.

Berpikir komputasional yang erat hubungannya dengan pemrograman akan dikemas menjadi bab dalam buku ini:

- **Bab 2**, Strategi Algoritmik dan Pemrograman, terdiri atas serangkaian aktivitas-aktivitas berkesinambungan yang merekatkan **Berpikir Komputasional**, **Algoritma Pemrograman**, **Analisis Data**, dan **Praktik Lintas Bidang**. Melalui pengemasan ini, pembelajaran diharapkan akan lebih bermakna karena kalian akan memahami bahwa berpikir komputasional dapat menjadi dasar dari konstruksi algoritma yang dipilih untuk diprogram dalam sebuah bahasa pemrograman. Ketika membuat program tersebut, kalian harus membuat model komputasi yang didasari oleh suatu bidang pengetahuan lain. Pada buku ini dipilih Kimia, Fisika, dan Biologi.
- **Bab 3**, intinya adalah **Dampak Sosial Informatika**, yang akan dipakai sebagai wahana membentuk berpikir kritis yang lebih tajam terhadap sebuah kasus sosial yang timbul akibat penerapan teknologi, khususnya teknologi informasi.
- **Bab 4**, tentang **Jaringan Komputer dan Internet** diperdalam konsepnya, namun tidak harus dipraktikkan karena aspek praktis jaringan komputer merupakan ranah pendidikan vokasional.
- **Bab 5**, **Praktik Lintas Bidang** dirancang untuk pengembangan aplikasi mobile yang dijalankan dengan smartphone atau gawai, dan berbasis Artificial Intelligence (AI) atau Kecerdasan Artifisial yang merupakan keniscayaan di zaman ini. Bukankah kalian sehari-hari banyak menggunakan HP dan aplikasinya? Jangan khawatir tentang AI karena di tingkatan Kelas XI, detail implementasi AI sudah dikemas menjadi sebuah modul "black box" yang tersedia dan dapat dipakai tanpa perlu mengimplementasi detailnya.

- **Bab 6**, merupakan sebuah proyek **Analisis Data** tentang sebuah isu yang merupakan salah satu isu internasional saat ini. Proyek ini diharapkan dapat memberikan gambaran tentang peran analisis data dalam membentuk manusia memahami apa yang terjadi dengan lebih akurat untuk mengambil keputusan yang tepat.

Tentunya, kalian mengenali elemen-elemen Informatika yang menjadi topik utama, dan cermatilah bahwa hampir semua praktik serta latihan dilakukan dengan menggunakan **Sistem Komputer**. Walaupun sistem komputer tidak dipelajari lagi secara khusus di kelas ini, namun tetap dibutuhkan dalam melakukan semua kegiatan pembelajaran.

Informatika terdiri atas konsep dan praktik yang dikemas dalam aktivitas pembelajaran dengan harapan akan menjadi pengalaman belajar yang menyenangkan, bermakna, dan berkesan agar dapat diterapkan dalam memecahkan persoalan kehidupan sehari-hari. Contoh kasus dalam buku ini menggunakan kasus yang umum. Kalian juga perlu bersiap diri jika guru menggantinya dengan kasus yang lebih dekat dengan konteks sehari-hari kalian, namun prinsipnya tetap setara. Pembelajaran Kelas XI membuka sekat-sekat dan merekatkan elemen Informatika, serta mengintegrasikan dengan bidang lain secara utuh dalam aktivitas berpikir, berkarya, berkolaborasi membangun produk komputasional lintas bidang ilmu. Aspek berpikir dapat dan bahkan lebih banyak dilakukan tanpa menggunakan komputer, sedangkan berkarya memerlukan komputer dan perangkat teknologi lain untuk tingkatan Kelas XI ini.

Ayo, Kita Berdiskusi

Aktivitas Kelompok

Aktivitas IF-K11-02-U: Diskusi tentang Materi Informatika

Setelah membaca dan mempelajari bacaan di atas, dan mempelajari kembali buku Kelas VII sampai dengan Kelas X, tentunya kalian akan lebih memahami apa yang akan dipelajari di Kelas XI, dan perbedaannya terutama dengan Kelas X. Pemahaman yang telah dilakukan secara mandiri dan individual akan dibahas pada sebuah diskusi kelompok, dan hasil setiap kelompok akan disampaikan pada sebuah diskusi panel.

Pertanyaan Pemantik Diskusi:

1. Apakah kalian sudah mempelajari semua elemen Informatika? Tuliskan elemen-elemen yang telah kalian pelajari, berurutan:

   a. dari yang paling mudah sampai dengan paling sulit

   b. dari yang paling menarik sampai dengan paling membosankan

   c. dari yang paling aplikatif (dirasakan dalam kehidupan sehari-hari) sampai dengan yang hanya konseptual saja

   Jelaskan alasanmu memilih urutan tersebut!

2. Perhatikanlah gambar "rumah" Informatika. Menurut kalian, mengapa setiap elemen tidak digambar dengan simbol yang sama?

3. Apa yang terjadi jika fondasi sebuah rumah tidak kokoh atau bahkan dibangun tanpa fondasi? Kemungkinan apa yang terjadi, jika berpikir komputasional tidak kuat?

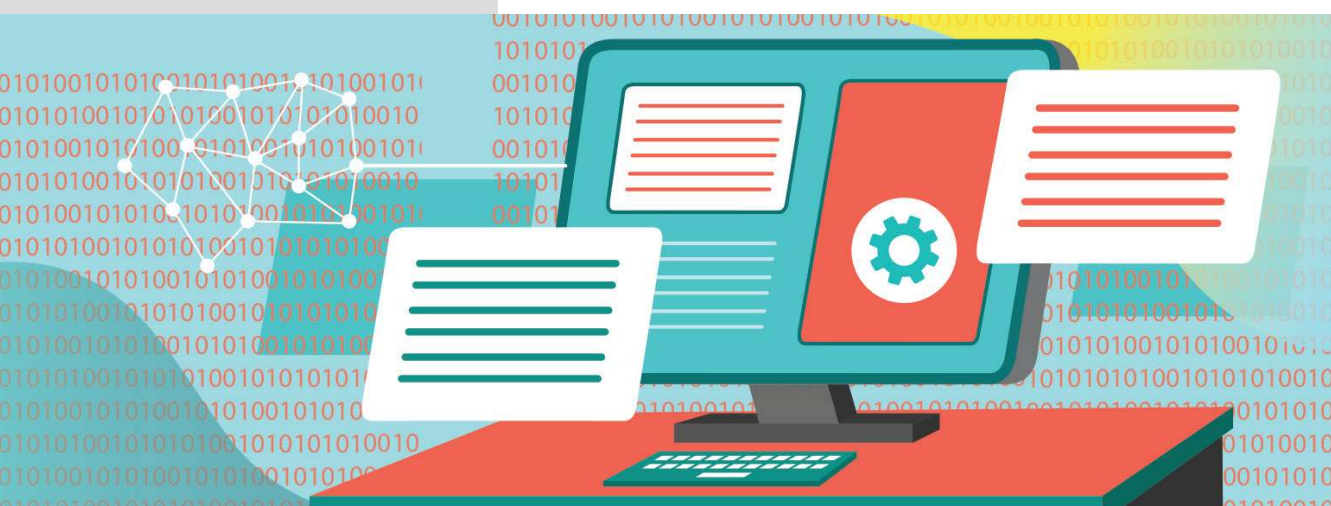
4. Apa yang terjadi jika pilar-pilar pengetahuan punya fondasi kokoh, tapi tidak diimplementasi menjadi karya berguna?
5. Apakah mungkin sebuah karya komputasional diwujudkan tanpa perangkat TIK?

Rangkumlah hasil diskusi kelompok dalam Lembar Kerja Kelompok yang disediakan oleh guru atau dalam bentuk bebas baik secara luring (lembar *flip chart paper*) atau secara daring (lembar kerja yang diisi dengan berkolaborasi secara daring), untuk menjadi bahan diskusi panel. Sajikan dalam bentuk visual yang mudah dipahami serta dapat membantu presentasi kelompok. Seluruh kelompok juga disarankan untuk membuat usulan ke guru mengenai materi Informatika yang masih memerlukan penguatan.

Ayo Renungkan

Setelah melakukan dua aktivitas sebelumnya mengenai materi Informatika yang akan dipelajari pada Kelas XI, tentu kalian sudah memahami tujuan dan harapan di akhir pembelajaran Informatika pada jenjang ini. Untuk selanjutnya cobalah deskripsikan rencana belajar dan hal yang ingin kalian capai selama menempuh Mata Pelajaran Informatika di Kelas XI yang telah kalian pilih dengan tanggung jawab serta kesadaran penuh. Kalian dapat menuliskannya dalam Lembar Refleksi pada Buku Kerja kalian. Apa yang kalian tulis akan menjadi catatan harapan yang dapat menjadi pengingat dan penambah semangat belajar. Mari menjalani proses belajar bermakna pada mata pelajaran yang telah kalian pilih!

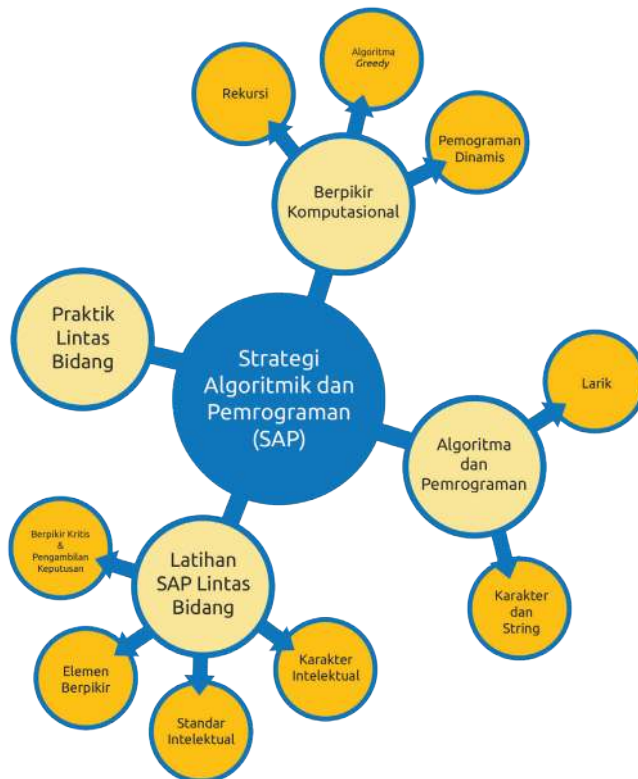
KEMENTERIAN PENDIDIKAN,
KEBUDAYAAN, RISET, DAN
TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas XI
Penulis : Mushtofa
Auzi Asfarian
Dean Apriana Ramadhan
ISBN: 978-602-244-861-7

2 Strategi Algoritmik dan Pemrograman

Tujuan Pembelajaran

Setelah mempelajari bab ini kalian diharapkan mampu menganalisis beberapa strategi algoritmik secara kritis untuk menyelesaikan suatu permasalahan dengan memberikan justifikasi efisiensi. Setelah itu, kalian akan menerapkan strategi algoritmik tersebut ke dalam suatu program yang lebih kompleks dari program yang telah kalian kerjakan sewaktu Kelas X. Kalian akan belajar bergotong royong dalam mengembangkan dan menguji program tersebut untuk menyelesaikan problem yang berkaitan dengan mata pelajaran lain yang telah kalian pelajari.

Pertanyaan Pemantik

Pernahkah kalian berpikir bagaimana program yang kita gunakan dalam kehidupan sehari-hari dikembangkan? Bagaimana para pemrogram menghasilkan program yang sangat kompleks yang kalian gunakan di kehidupan sehari-hari? Bagaimana program tersebut dirancang dan diimplementasikan hingga dapat melayani ribuan hingga jutaan pengguna dengan efisien sehingga program tetap terasa nyaman saat digunakan?

Peta Konsep

Gambar 2.1 Peta Konsep Strategi Algoritmik dan Pemrograman

Apersepsi

Dalam kehidupan ini, kalian pastilah telah menggunakan berbagai program untuk mempermudah aktivitas di berbagai bidang, seperti komunikasi, kesehatan, sosial, transportasi, hingga kebutuhan pokok sehari-hari. Akan tetapi, program yang bermanfaat tidak sekedar dapat dijalankan, namun bagaimana program tersebut dapat berfungsi dengan benar dan efisien sehingga dapat digunakan oleh manusia dengan nyaman. Untuk menghasilkan program demikian, diperlukan suatu kemampuan menyusun strategi algoritmik dan pemrograman.

Kata Kunci

Strategi algoritmik, pemrograman, penyelesaian masalah.

A. Proses Pemrograman

Pada buku Informatika SMA Kelas X, kalian telah mempelajari proses menulis kode program atau yang kita sebut sebagai *coding*. Pada Kelas XI, kalian akan mempelajari kegiatan yang lebih kompleks dari *coding* yang kita sebut sebagai pemrograman. Pemrograman menurut buku “Oxford Dictionary of Computer Science” adalah seluruh aktivitas teknis yang dilakukan untuk menghasilkan suatu program, termasuk analisis kebutuhan dan seluruh langkah desain dan implementasi suatu program.

Program dapat menjadi solusi dari suatu permasalahan. Untuk menghasilkan program yang benar dan dapat membantu manusia dalam melakukan tugasnya, ada empat langkah yang dilakukan pada saat melakukan pemrograman, yaitu:

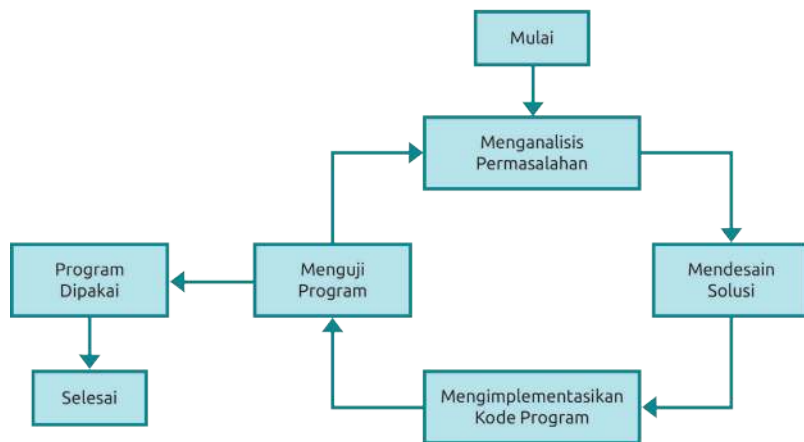
1. **Menganalisis permasalahan (*Analyzing*)**: pemrogram menganalisis suatu kebutuhan atau keadaan saat ini untuk menghasilkan definisi permasalahan yang perlu

diselesaikan dengan program. Permasalahan yang dianalisis bisa berupa masalah yang baru atau penyempurnaan dari solusi yang sudah ada. Kemampuan berpikir komputasional digunakan untuk mencari abstraksi dari permasalahan yang akan diselesaikan. Permasalahan yang kompleks dapat didekomposisi ke beberapa masalah yang lebih kecil, namun saling berkaitan. Hingga akhirnya, pemrogram akan mengenali pola permasalahan tersebut sebagai sebuah variasi dari *problem* generik. Tahap ini menghasilkan pernyataan masalah (*problem statement*) yang menjelaskan masukan (*input*), keluaran (*output*), serta batasan-batasan (*constraint*) dari program yang akan dibuat.

2. **Mendesain solusi (*Problem Solving*):** dari pernyataan masalah ini, pemrogram merencanakan strategi untuk menghasilkan keluaran berdasarkan masukan yang diterima. Pemrogram tidak harus merencanakan solusi dari awal. Mereka dapat menggunakan solusi atau potongan solusi yang sudah ada dari permasalahan yang telah diselesaikan sebelumnya. Bahkan, jika problem generik telah ditemukan, pemrogram dapat memodifikasi algoritma generik agar sesuai dengan permasalahan. Proses ini akan menghasilkan algoritma berupa narasi, *pseudocode*, atau diagram alir. Pada tahap ini, pemrogram juga akan mengevaluasi algoritma yang dibuat untuk memenuhi batasan dari permasalahan. Misalnya, apakah program dapat bekerja dengan cepat (kurang dari 1 detik) saat diberikan masukan yang berukuran besar, karena menunggu membuat pengguna tidak nyaman.
3. **Mengimplementasikan solusi dalam bentuk program (*Coding*):** Pada tahap ini, pemrogram akan menulis kode program untuk menjalankan solusi yang telah direncanakan sebelumnya dengan menggunakan suatu bahasa pemrograman. Memilih bahasa pemrograman menjadi pertimbangan di tahap ini. Selain itu, mengubah

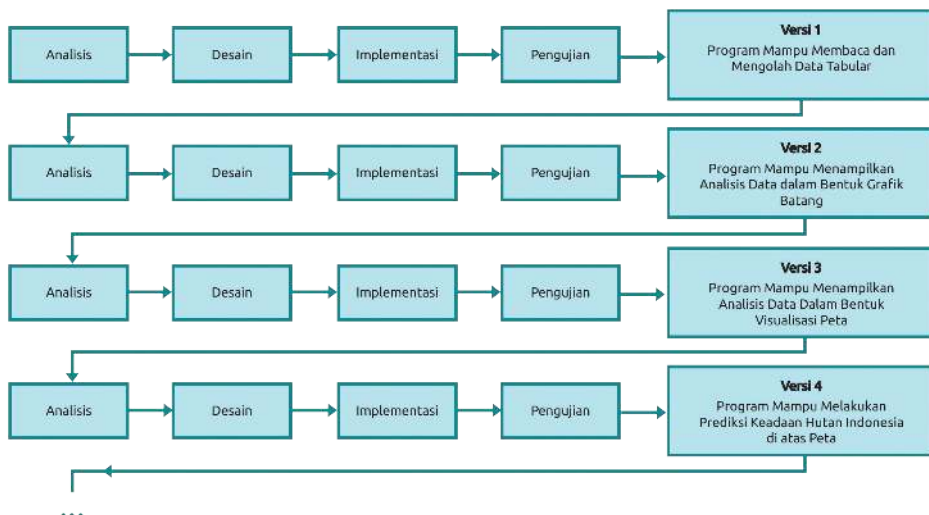
algoritma menjadi kode program juga melibatkan banyak pertimbangan teknis (misalnya: tipe data, struktur kontrol yang digunakan, dan lain-lain).

4. **Menguji program (*Testing*)**: Setelah program dapat dijalankan, program tersebut harus diuji untuk memastikan program berjalan dengan benar, sesuai dengan batasan-batasan yang diberikan. Pengujian dapat dilakukan dengan menggunakan berbagai strategi pengujian. Salah satu yang telah kalian pelajari di Kelas X adalah menguji program menggunakan kasus uji (*test case*) yang dibuat sedemikian rupa sehingga mewakili seluruh kemungkinan masukan dari program.

Keempat tahap di atas seringkali tidak dilakukan satu kali, tapi berkali-kali sehingga membentuk suatu siklus pemrograman (lihat Gambar 2.2).

Gambar 2.2 Siklus Pemrograman

Dalam latihan di buku Informatika Kelas X, kalian mungkin merasakan bahwa program yang kalian buat masih jauh lebih sederhana dibandingkan dengan program yang biasa kalian gunakan dalam kehidupan sehari-hari. Hal tersebut sangatlah wajar, karena program-program yang kompleks dengan tampilan visual dan interaktif yang banyak dipakai saat ini telah melewati tahapan pengembangan yang panjang dan

telah melalui beberapa versi. Apabila kalian menggunakan sebuah program yang disediakan secara daring (misalnya untuk berbelanja) dan rutin dalam waktu lama, kalian akan merasakan bahwa program tersebut senantiasa mengalami perubahan dan perbaikan atau penambahan fitur secara terus menerus. Hal ini menjadi bagian dari "siklus kehidupan" suatu program agar terus relevan dengan kebutuhan penggunanya.

Pendekatan siklus ini juga bermanfaat jika kalian ingin membuat sebuah program yang kompleks. Kalian dapat menetapkan suatu bentuk akhir dari program yang akan kalian buat dan mendekomposisinya ke dalam program-program antara yang lebih sederhana. Misalnya, kalian ingin membuat program yang mampu memprediksi kondisi hutan Indonesia di masa depan. Untuk membuat program yang kompleks tadi, kalian dapat mulai dengan membuat program yang dapat membaca dan mengolah data hutan dalam bentuk tabel terlebih dahulu, kemudian menambahkan analisis data sederhana dalam bentuk grafik batang, kemudian mengubahnya menjadi visualisasi berbentuk peta, baru akhirnya menambah kemampuan prediksi keadaan hutan (lihat Gambar 2.3).

Gambar 2.3 Contoh Pembuatan suatu Program Prediksi Keadaan Hutan Indonesia yang Dibangun secara Iteratif

Praktik Baik Pemrograman

Kalian perlu mengumpulkan dengan rapi hasil pekerjaan dari langkah di atas. Misalnya, coret-coretan, algoritma yang dibuat, kode sumber, serta proses pengujian yang dilakukan. Hal ini disebut dokumentasi program. Dokumentasi berguna bagi kalian maupun orang lain untuk memahami kode program dengan cepat. Dokumentasi dapat ditulis baik di dalam kode program maupun di dokumen terpisah (untuk program yang kompleks). Dokumentasi yang ditulis menyatu dengan kode dapat ditulis menggunakan komentar, namun komentar tidak perlu menjelaskan setiap baris kode, cukup menjelaskan abstraksi/ide dari kode. Ingatlah bahwa kode program yang ditulis dengan baik sudah pasti akan mudah dipahami.

Contoh Kasus Siklus Pemrograman

Selanjutnya mari kita coba lakukan keempat langkah tersebut untuk membuat sebuah program yang dapat membantu seseorang mengidentifikasi jenis segitiga berdasarkan panjang ketiga sisinya. Seperti kita ketahui, ada beberapa jenis segitiga berdasarkan kondisi panjang masing-masing sisinya, misalnya segitiga sama sisi, segitiga sama kaki, atau pun segitiga sembarang. Bayangkan sebuah program yang dapat menentukan apakah sebuah segitiga termasuk segitiga sama kaki, sama sisi, atau bukan keduanya, jika diberikan data berupa ketiga panjang sisi-sisi dari segitiga yang dimaksud. Misalnya jika diberikan masukan berupa panjang sisi-sisi: 2, 2 dan 3, maka program tersebut tentunya harus menghasilkan keluaran “segitiga sama kaki”.

1. Mendefinisikan Permasalahan

Deskripsi di atas sangat abstrak dan luas. Oleh karena itu, sebagai pemrogram kita perlu mendefinisikan ruang lingkup dari permasalahan yang diberikan. Pada pengembangan perangkat lunak, ruang lingkup tersebut disebut sebagai kebutuhan (requirement). Definisi ini perlu dibuat dengan baik karena pengecekan kebenaran program yang kita buat akan bergantung pada definisi permasalahan yang telah dibuat. Untuk menggali

requirement, kalian dapat bertanya kepada pembuat soal, kepada guru, atau kepada orang-orang yang nantinya akan menggunakan program kalian.

Misalnya pada deskripsi mengidentifikasi jenis segitiga di atas, kita perlu mencari jawaban dari beberapa pertanyaan berikut: (a) Bagaimana caranya mengidentifikasi apakah data panjang sisi-sisi tersebut menunjukkan sisi sebuah segitiga? (karena bisa jadi tidak ada segitiga yang memiliki panjang sisi-sisi sebagaimana data yang diberikan) (b) Bagaimana menentukan jenis segitiga dari panjang sisi-sisinya? (c) Jenis segitiga apa saja yang harus kita identifikasi (misalnya: apakah kita harus bisa mengidentifikasi segitiga siku-siku?) (d) Batasan seperti apa yang harus dipenuhi oleh data masukan? (e) keluaran seperti apa yang harus diberikan oleh program? dan seterusnya. Setelah pertanyaan-pertanyaan tersebut terjawab, kalian dapat membuat definisi permasalahan yang lebih formal. Contoh berikut dapat menjadi salah satu cara untuk mendefinisikan permasalahan ini setelah kalian mendapatkan jawaban dari pertanyaan-pertanyaan di atas.

Mengidentifikasi Jenis Segitiga

Deskripsi:

Diberikan tiga buah bilangan bulat yang berada pada rentang [1, 1000] yang merupakan panjang sisi dari sebuah segitiga. Identifikasi apakah ketiga sisi tersebut membentuk segitiga sama sisi, atau segitiga sama kaki, segitiga sembarang, atau tidak bisa membentuk segitiga!

Masukan:

Masukan terdiri atas tiga bilangan bulat a, b, dan c yang merupakan panjang masing-masing sisi segitiga pada rentang [1, 1000]

Proses:

Dari masukan yang diberikan, kita harus menentukan terlebih dahulu, apakah ada segitiga dengan panjang sisi-sisi a, b dan c. Jika tidak ada, maka program dapat berhenti dan melaporkan

bahwa data masukan tidak menunjukkan sisi-sisi sebuah segitiga. Jika ternyata a, b dan c merupakan sisi-sisi sebuah segitiga, maka program harus menentukan jenis segitiga apakah yang memiliki panjang sisi-sisi a, b dan c tersebut.

Keluaran:

Keluaran berupa teks sebagai berikut ini:

- "Segitiga Sama Sisi" jika masukan berupa segitiga sama sisi.
- "Segitiga Sama Kaki" jika masukan berupa segitiga sama kaki.
- "Segitiga Sembarang" jika masukan berupa segitiga sembarang.
- "Bukan Segitiga" jika masukan bukan berupa segitiga.

2. Merancang Solusi

Setelah definisi persoalan (masukan, proses, keluaran) dibuat seperti di atas. Selanjutnya kalian akan mencoba menyusun algoritma untuk menyelesaikan permasalahan tersebut. Pertama-tama, untuk menentukan apakah ada segitiga yang memiliki sisi-sisi dengan panjang sesuai masukan, yaitu a, b dan c, maka nilai-nilai tersebut haruslah memenuhi aturan "Teorema Pertidaksamaan Segitiga" berikut:

"Untuk semua segitiga dengan panjang a, b dan c, maka haruslah berlaku $a + b > c$"

Jika kita terjemahkan teorema di atas untuk sebarang masukan a, b, dan c, kita harus memeriksa 3 buah kondisi berikut:

- $a + b > c$
- $a + c > b$
- $b + c > a$

Ketiga kondisi tersebut harus terpenuhi, agar a, b dan c dapat membentuk segitiga. Dengan kata lain, jika setidaknya salah satu kondisi tersebut tidak dipenuhi, maka ketiga sisi tersebut tidak bisa membentuk segitiga.Jika hal ini terjadi, program dapat melaporkan hasil ini dan kemudian langsung berhenti.

Jika tidak, berarti a, b dan c memang benar merupakan sisi-sisi sebuah segitiga, dan program dapat mengidentifikasi jenis segitiga yang sesuai.

Selanjutnya, untuk mengidentifikasi jenis segitiga, kita dapat merancang aturan-aturan sebagai berikut:

- Jika tiga panjang sisi sama, maka segitiga tersebut adalah sama sisi;
- Jika hanya dua panjang sisi sama, maka segitiga tersebut sama kaki;
- Jika ketiga panjang sisi berbeda, maka segitiga tersebut adalah segitiga sembarang.

3. Mengimplementasikan Solusi dalam Bentuk Program (*Coding*)

Selanjutnya, kalian dapat mengimplementasikan algoritma tersebut ke dalam suatu program komputer. Misalnya, kita dapat mencoba implementasikan dalam bahasa pemrograman C. Tentunya, kalian juga dapat menuliskan algoritma di atas dalam bahasa pemrograman lainnya yang telah kalian pelajari seperti Python, atau bahkan menggunakan pemrograman blok.

Catatan: Saat menuliskan kode yang ada pada buku ini, kalian tidak perlu menuliskan nomor baris pada program kalian. Nomor baris pada buku ini ditambahkan untuk mempermudah kalian dalam membaca kode sumber.

```
1:  /* Program untuk Menentukan Jenis Segitiga */
2:  #include <stdio.h>
3:  int main()
4:  {
5:      int a, b, c;
6:      scanf("%d %d %d", &a, &b, &c);
7:
8:      if ((a >= b + c) || (b >= a + c) || (c >= a + b))
9:          printf("Bukan Segitiga");
10:     else
11:         if ((a == b) && (b == c))
12:             printf("Segitiga Sama Sisi");
13:         else if ((a == b) || (b == c) || (a == c))
14:             printf("Segitiga Sama Kaki");
15:         else
16:             printf("Segitiga Sembarang");
17:     return 0;
18: }
```

4. Menguji Program

Setelah program selesai, kita perlu menguji kebenaran program tersebut. Ada banyak metode formal untuk menguji suatu program, namun yang akan dicontohkan adalah menguji program menggunakan kasus uji seperti yang telah diberikan di Kelas X. Misalkan, kita akan menguji kode program di atas dengan menggunakan pasangan kasus uji pada tabel berikut:

Tabel 2.1 Lembar kerja untuk menguji Program Jenis Segitiga

No	Masukan	Keluaran Seharusnya	Keluaran Program	Hasil Benar?
1	5 5 5	Segitiga Sama Sisi		
2	1000 1000 500	Segitiga Sama Kaki		
3	100 200 500	Bukan Segitiga		
4	100 400 200	Segitiga Sembarang		

Apabila terdapat kasus uji yang tidak dapat menghasilkan keluaran yang benar oleh program, lakukan analisis kembali

untuk mengidentifikasi kesalahan tersebut dan memperbaiki program kalian hingga benar.

Ini Informatika

Dari ilustrasi di atas, kalian telah merasakan bahwa dalam membuat program, kalian perlu memahami permasalahan yang kalian kerjakan. Semakin kompleks suatu permasalahan yang harus diselesaikan, atau semakin banyak fitur yang harus diimplementasikan, program juga akan menjadi semakin kompleks dan mungkin menjadi panjang (lebih banyak instruksinya). Oleh karena itu, tim yang membuat program akan terdiri atas banyak orang dengan keahlian dan spesialisasi yang berbeda. Di antara mereka juga ada orang-orang yang ahli dan paham dengan permasalahan yang diselesaikan.

Ayo Berlatih!

Aktivitas Individu

Aktivitas SAP-K11-01-U: Mengamati Evolusi Program

Deskripsi Tugas

Pada bagian ini telah dijelaskan bahwa suatu program dikembangkan setahap demi setahap dalam suatu siklus. Hal ini juga berlaku pada program, baik berbentuk aplikasi atau web yang kalian gunakan. Sekarang pilihlah satu aplikasi atau web yang sering kalian gunakan, kemudian buatlah infografis linimasa (timeline) sederhana yang menunjukkan proses perubahan program seiring dengan berjalannya waktu. Kalian dapat melakukan pencarian untuk menemukan versi lama dari program. Untuk web, kalian dapat menggunakan aplikasi web seperti web.archive.org yang dapat menyimpan versi lama dari web secara periodik.

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah kalian telah dapat memahami setiap proses dalam pemrograman?
2. Tahap apa yang menurut kalian paling menarik untuk dilakukan?
3. Tahap apa yang menurut kalian paling sulit untuk dilakukan?
4. Pelajaran paling berkesan apa yang Kalian dapatkan dari latihan ini?

B. Berpikir Komputasional

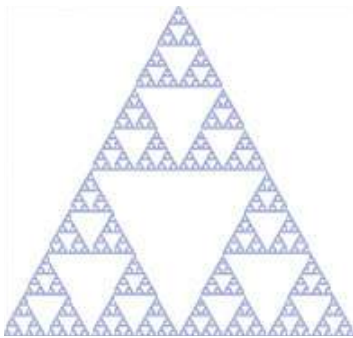
Berpikir komputasional ini merupakan suatu kerangka dan proses berpikir yang mencakup perangkat keras, perangkat lunak, dan menalar (*reasoning*) mengenai sistem dan persoalan. Moda berpikir (*thinking mode*) ini didukung dan dilengkapi dengan pengetahuan teoritis dan praktis, serta teknik untuk menganalisis, memodelkan dan menyelesaikan persoalan. Berikut ini pembahasan beberapa konsep dan strategi berpikir komputasional yang biasa digunakan dalam menyelesaikan persoalan komputasi.

1. Rekursi

Pada bagian ini kalian akan mempelajari tentang konsep rekursi dan beberapa contoh permasalahan yang dapat diselesaikan dengan menggunakan konsep tersebut, salah satunya adalah barisan Fibonacci.

Gambar 2.4 Ilustrasi dari Rekursi

Suatu masalah dapat didekomposisi menjadi permasalahan yang serupa, namun ukurannya lebih kecil. Saat kita diminta untuk memindahkan satu kardus buku yang sangat berat dan tidak dapat kita angkat, kita akan membagi kardus tersebut ke dalam beberapa kardus yang lebih ringan sehingga pekerjaan tersebut menjadi lebih mudah untuk dikerjakan. Ketika menghitung suatu nilai faktorial, kita pun harus menghitung nilai faktorial yang lebih kecil. Misalnya, ketika menghitung 10 faktorial, kita juga harus menyelesaikan 1 faktorial, 2 faktorial, hingga 9 faktorial terlebih dahulu.

Secara alami, terdapat banyak permasalahan yang dapat dimodelkan dengan lebih mudah menggunakan konsep rekursif ini. Pada bagian ini, kalian akan mempelajari konsep dasar rekursi yang akan sangat berguna untuk melakukan dekomposisi pada suatu permasalahan besar dalam bentuk permasalahan yang lebih kecil dan lebih mudah untuk diselesaikan. Rekursi didefinisikan sebagai "sesuatu" yang mengandung "sesuatu" itu sendiri. Dapatkah kalian melihat rekursi dalam gambar-gambar sebagai berikut (lihat Gambar 2.5)

Segi tiga Sierpinski

Boneka bersarang dari Rusia

Gambar 2.5 Beberapa contoh Rekursi di Kehidupan Sehari-hari

Dalam pembahasan kali ini, kita akan membahas fungsi/barisan rekurensi (recurrence) yaitu fungsi/barisan dimana nilai dari

fungsi atau barisan tersebut ditentukan/tergantung dari nilai fungsi/barisan itu sendiri secara rekursif, pada urutan nilai-nilai sebelumnya. Misalnya, kita memiliki sebuah barisan ai,i=1,2,...,n sebagai berikut:

$$\{a_i\} = 1, 3, 5, 7, ...$$

Dimana nilai pertama dari barisan (a_1) adalah 1, dan kemudian nilai-nilai berikutnya dalam barisan tersebut dihitung dengan cara menambahkan nilai 2 kepada nilai barisan sebelumnya. Kita dapat menuliskan dalam notasi rekursif sebagai berikut:

$$a_i \begin{cases} 1, \text{ jika i} = 1 \\ a_{i-2} + 2, \quad \text{jika i} > 1 \end{cases}$$

Pada definisi sebuah barisan/fungsi rekursif, selalu ada minimal dua hal yang harus ditentukan, yaitu:

- **Basis**: menunjukkan dasar/nilai awal dari fungsi/barisan tersebut. Misalnya, pada contoh di atas, $a_1=1$
- **Rekursi**: menunjukkan hubungan antara nilai dari fungsi/barisan tersebut dengan nilai-nilai sebelumnya yang telah diketahui. Misalnya, pada contoh di atas: $a_i=a_{i-1}+2$, jika $i > 1$.

Sebuah fungsi/barisan rekursif bisa jadi ditentukan dari tidak hanya satu buah nilai sebelumnya saja, tetapi dapat juga dari 2, 3, ... dan seterusnya, nilai sebelumnya. Sebagai contoh sebuah barisan dapat didefinisikan sebagai berikut:

$$a_i = \begin{cases} 1, \text{ jika i} = 1 \textit{ atau } \text{i} = 2 \\ a_{i-1} + a_{i-2} \quad \text{jika i} > 2 \end{cases}$$

Barisan ini dimulai dengan nilai 1, kemudian untuk menentukan nilai berikutnya, kita hitung dengan cara menjumlahkan dua nilai sebelumnya pada barisan tersebut, sehingga didapatkan barisan sebagai berikut:

$$\{a_i = 1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

Barisan di atas biasa disebut sebagai barisan Fibonacci, karena dipopulerkan oleh seorang matematikawan Italia bernama Fibonacci (nama lengkap Leonardo Bonacci, 1170 – 1250 M). Perlu diperhatikan bahwa karena pada bagian rekursi, kita memerlukan dua nilai terakhir, pada bagian basis, kita perlu mendefinisikan dua nilai pertama dari barisan tersebut. Secara umum, banyaknya nilai yang harus didefinisikan pada bagian basis ditentukan oleh banyaknya suku barisan yang diperlukan pada bagian definisi rekursi.

Ayo Berlatih!

Aktivitas Individu

Aktivitas SAP-K11-02-U: Memahami Relasi Rekurensi

Deskripsi Tugas

Relasi rekurensi (*recurrence relation*) adalah sebuah tipe relasi matematis dimana definisi dari sebuah fungsi atau barisan dinyatakan secara rekursif, artinya merujuk pada fungsi atau barisan itu sendiri. Pada bagian ini, kalian akan berlatih untuk memahami definisi relasi rekurensi dan bagaimana menerapkannya, serta membuat definisi rekursif dari sebuah permasalahan.

1. Tentukan suku ke-10 dari barisan yang didefinisikan sebagai berikut:

$$a_i = \begin{cases} 1 \text{ jika i} = 1 \text{ atau i} - 2 \\ 2a_{i-1} + 2a_{i-2} \text{ jika i} > 2 \end{cases}$$

   atau dengan kata lain, barisan tersebut dimulai dengan nilai 1, 1, kemudian untuk menghitung suku berikutnya, kita jumlahkan antara suku sebelumnya dengan dua kali dari suku sebelum suku sebelumnya.

2. Faktorial dari sebuah bilangan bulat n ≥ 1, ditulis sebagai n!, didefinisikan sebagai sebuah nilai yang dihitung dengan mengalikan semua bulat dari 1 sampai dengan n.

Sebagai contoh, faktorial dari 5 adalah $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$. Buatlah sebuah definisi rekursif untuk menghitung nilai n!.

Ayo Berlatih!

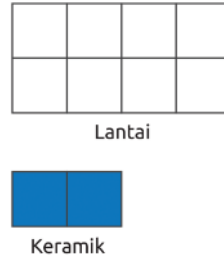
Aktivitas Individu

Aktivitas SAP-K11-03-U: Menerapkan Konsep Rekursi

Deskripsi Tugas

Selesaikanlah dua problem berikut dengan menerapkan konsep rekursi yang telah kalian pelajari. Setelah mengerjakan problem tersebut, diskusikanlah solusi kalian dengan teman.

Permasalahan 1: Memasang Keramik

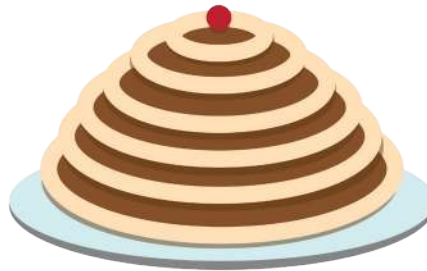
Terdapat sebuah lantai yang berukuran $2 \times N$. Pada lantai tersebut, ingin dipasang N buah keramik, yang masing-masing berukuran 1×2 (perhatikan ilustrasi pada gambar 2.6).

Gambar 2.6 Ilustrasi memasang keramik

Setiap keramik dapat dipasang secara mendatar (horizontal) ataupun secara tegak (vertikal). Sebagai contoh, jika N=4, maka akan ada 5 buah cara berbeda memasang 4 keramik, yaitu:

- Semua keramik dipasang secara horizontal.
- Semua keramik dipasang secara vertikal.
- Paling kiri ada 2 keramik vertikal, sisanya horizontal.
- Paling kiri ada 2 keramik horisontal, sisanya vertikal.

- Paling kiri dan paling kanan vertikal, sisanya horizontal.
- Tentukan ada berapa cara memasang keramik untuk N=8?

Permasalahan 2: Menumpuk Panekuk

Budi memiliki setumpuk panekuk yang memiliki ukuran dari besar sampai kecil. Panekuk-panekuk tersebut ditumpuk di atas sebuah piring, dengan aturan bahwa panekuk yang besar harus berada di bawah panekuk yang lebih kecil.

Gambar 2.7 Kue Panekuk

Budi ingin memindahkan panekuk ini dari satu piring ke piring lainnya, namun dalam prosesnya ia tetap ingin mengikuti aturan bahwa panekuk yang besar harus selalu berada di bawah panekuk yang lebih kecil. Selain itu, Budi juga hanya boleh memindahkan satu buah panekuk saja, pada satu waktu tertentu, dari satu piring ke pring lainnya. Budi menyadari bahwa ia memerlukan sebuah piring tambahan untuk dapat melakukan perpindahan ini. Jika piring asal panekuk diberi nama A, piring tujuan diberi nama C, maka Budi akan menyiapkan sebuah piring bantuan sebagai tempat sementara, yang diberi nama piring B.

Budi ingin mengetahui, berapa banyak langkah pemindahan panekuk yang harus ia lakukan, untuk dapat memindahkan semua panekuk yang dimilikinya, dari piring A ke piring C (dengan mungkin menggunakan piring B sebagai tempat sementara). Misalnya, jika Budi memiliki 2 buah panekuk saja (diberi nama panekuk 1 dan, dimana panekuk 1 berukuran lebih kecil dari panekuk 2), maka ia akan membutuhkan minimal 3 langkah pemindahan:

1. Pindahkan panekuk 1 dari piring A ke piring B
2. Pindahkan panekuk 2 dari piring A ke piring C
3. Pindahkan panekuk 1 dari piring B ke piring C

Berapakah jumlah langkah minimal yang diperlukan apabila Budi memiliki 6 buah panekuk?

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah kalian dapat menentukan nilai dari suatu barisan/ fungsi, apabila diberikan nilai awal dan definisi relasi rekurensi-nya?
2. Apakah kalian sudah berhasil memodelkan contoh masalah pada latihan-latihan di atas dalam bentuk relasi rekurensi?
3. Apakah kalian merasa ada masalah lain yang serupa (dalam kehidupan sehari-hari) yang dapat dimodelkan dalam bentuk relasi rekurensi tadi?
4. Pelajaran paling berkesan apa yang kalian dapatkan dari aktivitas latihan ini?

2. Algoritma Greedy

Greedy secara harfiah berarti rakus atau tamak. Meskipun dalam pengertian sehari-hari, kata "rakus" dan "tamak" memiliki konotasi negatif, namun dalam konteks Informatika, kita mengartikan *greedy* dalam konteks sebagai sebuah strategi penyelesaian masalah yang dapat berguna dalam merancang sebuah algoritma atau solusi bagi sebuah permasalahan komputasional. Oleh karena itu, diharapkan tidak ada konotasi negatif pada kata *greedy* dalam konteks ini.

Teknik *greedy* adalah salah satu teknik penyelesaian masalah yang biasa digunakan untuk menyelesaikan permasalahan optimasi. Permasalahan optimasi berarti kita ingin menghitung sebuah hasil yang terbaik dari sebuah proses tertentu. Terbaik disini dapat berarti nilai yang paling kecil ataupun paling besar, tergantung dari jenis permasalahannya. Dalam menyelesaikan permasalahan optimasi seperti ini, algoritma *greedy* akan menerapkan prinsip "mengambil serangkaian langkah terbaik pada setiap saat".

Contoh 1: Membawa Ikan 1

Budi ingin membawa beberapa ekor ikan yang sudah tersimpan dalam kantong-kantong plastik untuk diangkut di dalam mobilnya. Terdapat 8 buah kantong dengan yang berisi masing-masing 3, 5, 2, 8, 4, 6, 6, dan 3 ekor ikan. Namun sayangnya, mobilnya hanya mampu membawah 4 buah kantong. Kantong-kantong manakah yang harus dibawa oleh Budi agar jumlah ikan yang dibawanya sebanyak mungkin?

Jawab:

Untuk dapat membawa sebanyak mungkin ikan, Budi harus memilih kantong-kantong dengan sebanyak mungkin ikan. Oleh karena itu, algoritma *greedy* dapat diterapkan disini, dengan cara kita mengambil kantong mulai dari yang berisi ikan paling banyak terlebih dahulu, sampai didapatkan 4 buah kantong. Dengan demikian, kita harus mengurutkan kantong-kantong terlebih dahulu mulai dari yang paling banyak ikannya, sampai dengan yang paling sedikit, sehingga urutannya menjadi: 8, 6, 6, 5, 4, 3, 3, 2. Jika kita ambil 4 buah kantong pertama, maka total banyaknya ikan yang dapat dibawa adalah $8 + 6 + 6 + 5 = 25$ ekor ikan. Tentunya tidak ada pilihan 4 kantong yang akan menghasilkan total banyaknya ikan lebih dari 25 ekor.

Contoh 2: Membawa Ikan 2

Kali ini, Budi harus membawa sedikitnya 15 ekor ikan. Tentukan jumlah kantong terkecil yang harus dibawa oleh Budi, agar terdapat minimal 15 ekor ikan yang terbawa!

Jawab:

Sama seperti pada permasalahan sebelumnya, kita dapat menerapkan algoritma greedy untuk menyelesaikan permasalahan ini. Dalam hal ini, untuk memperkecil banyaknya kantong yang harus dibawa, maka kita juga selalu memilih kantong dengan jumlah ikan terbanyak terlebih dahulu. Jika kita memilih kantong dengan jumlah ikan = 8 dan 6, maka kita sudah memiliki 14 ekor ikan. Selanjutnya, kita hanya perlu mengambil 1 kantong lagi (yang mana saja) agar total jumlah ikan menjadi lebih dari 15. Oleh karena itu, jawaban yang diinginkan adalah 3 buah kantong. Jelas bahwa tidak ada pilihan yang memungkinkan kita mendapatkan 15 ekor ikan dengan 2 atau kurang kantong.

Pada kedua contoh di atas, terdapat satu langkah yang penting yang biasa diterapkan pada penyelesaian masalah secara *greedy*, yaitu proses mengurutkan sebuah data agar menjadi terurut (mungkin dari kecil ke besar, atau sebaliknya), agar kemudian kita dapat melakukan serangkaian pengambilan langkah secara *greedy* pada data yang sudah terurut tersebut. Pola seperti ini umum digunakan pada penyelesaian permasalahan secara *greedy*.

Ayo Berlatih!

Aktivitas Kelompok

Aktivitas SAP-K11-04-U: Mengerjakan Pekerjaan Rumah (PR)

Deskripsi Tugas

Cici menerima 10 buah pekerjaan rumah (PR) yang harus ia kerjakan. Setelah melihat isi dari masing-masing PR, Cici memiliki perkiraan, berapa lama waktu yang diperlukan untuk mengerjakan masing-masing PR tersebut, seperti terlihat pada tabel di bawah.

No.	PR	Waktu pengerjaan (jam)
1.	A	1.5
2.	B	3
3.	C	1
4.	D	0,5
5.	E	4

No.	PR	Waktu pengerjaan (jam)
6.	F	1
7.	G	2,5
8.	H	1
9.	I	0,5
10.	J	2

Sayangnya, ia tidak punya banyak waktu untuk mengerjakan semua PR. Cici menghitung bahwa ia hanya punya waktu total = 8 jam sebelum semua PR tersebut harus dikumpulkan. Cici ingin menentukan, PR mana yang harus ia kerjakan terlebih dahulu, dengan pertimbangan bahwa setiap PR memiliki nilai yang sama besarnya (terhadap nilai akhir Cici). Bantulah Cici menentukan PR yang mana saja yang harus ia kerjakan dalam waktu maksimal 8 jam, untuk mendapatkan total nilai akhir yang sebesar-besarnya.

Ayo Berlatih!

Aktivitas Individu

Aktivitas SAP-K11-05-U: Mengunjungi Kebun Binatang

Deskripsi Tugas

Dina sedang bertamasya mengunjungi kebun binatang. Setiap hari, kebun binatang mengadakan beberapa pertunjukan atraksi hewan yang dapat ditonton oleh para pengunjung. Berikut adalah jadwal yang telah ditetapkan oleh pengelola kebun binatang.

No.	Waktu Mulai	Waktu Selesai	Pertunjukan/Atraksi Hewan
1.	9:15	10:30	Orang Utan
2.	8:00	9:30	Pinguin
3.	10:00	12:00	Harimau
4.	13:00	14:30	Beruang Madu
5.	11:00	12:30	Burung Pemangsa
6.	14:00	15:00	Buaya
7.	15:30	16:30	Panda
8.	16:00	17:00	Ular Piton
9.	15:00	15:30	Singa
10.	15:30	16:00	Anjing Laut

Tentunya dalam satu waktu tertentu, Dina hanya dapat menonton satu pertunjukan atraksi hewan. Dina ingin dapat melihat sebanyak-banyaknya pertunjukan dalam satu hari tersebut, dan ia tidak memiliki preferensi dalam melihat pertunjukan hewan (semuanya ia anggap sama menariknya). Tentukan, ada berapa banyak maksimal pertunjukan yang dapat ditonton oleh Dina?

Ayo Berlatih!

Aktivitas Individu

Aktivitas SAP-K11-06-U: Menukarkan Uang

Deskripsi Tugas

Dalam kehidupan sehari-hari, kita pasti sudah banyak terbiasa dengan perhitungan yang melibatkan uang. Misalnya, ketika Anda membeli sebuah barang/makanan, atau pun ingin membayar untuk sebuah jasa tertentu, kita seringkali menyiapkan sejumlah uang tertentu, sesuai dengan harga barang atau jasa tersebut. Selanjutnya, bagi penjual atau penyedia jasa, apabila mereka menerima uang pembayaran dengan jumlah total yang lebih besar dari harga yang ditetapkan, mereka pun juga harus menyiapkan uang kembalian sesuai dengan jumlah kelebihan pembayaran.

Di Indonesia, mata uang rupiah memiliki beberapa pecahan uang, mulai dari yang terkecil, 100 Rupiah, 200 Rupiah, 500 Rupiah, dst sampai dengan 100 000 Rupiah. Seandainya kita memiliki sejumlah pecahan uang, misalnya beberapa uang seribuan, dua ribuan, lima ribuan, sepuluh ribuan dan dua puluh ribuan. Jika kita ingin mendapatkan uang tepat sejumlah 38 ribu rupiah, maka kita dapat memilih beberapa cara, misalnya:

- 3 lembar sepuluh ribuan, ditambah 1 lembar lima ribuan, ditambah 2 lembar ribuan, ditambah 2 koin lima ratus, dengan total ada 8 buah lembaran uang/koin
- 1 lembar dua puluh ribuan, ditambah 1 lembar sepuluh ribuan ditambah 4 lembar dua ribuan, totalnya menjadi 6 lembaran uang
- 1 lembar dua puluh ribuan, ditambah 1 lembar sepuluh ribuan, ditambah 1 lembar lima ribuan ditambah 1 lembar dua ribuan, ditambah 1 lembar seribuan, dengan total ada 5 lembaran uang.

Jelas bahwa jumlah total lembaran yang dibutuhkan tergantung dari pemilihan pecahan uang yang kita gunakan. Nah, permasalahan yang mungkin kita tanyakan adalah: Bagaimana caranya memilih pecahan-pecahan uang yang akan digunakan sedemikian rupa, sehingga total lembaran yang diperlukan untuk menghasilkan suatu nilai uang tertentu menjadi sekecil mungkin?

Pada contoh di atas, dapat diperiksa bahwa untuk menghasilkan nilai uang sebesar tiga puluh delapan ribu rupiah dari pecahan-pecahan seribuan, dua ribuan, lima ribuan, sepuluh ribuan dan dua puluh ribuan, maka diperlukan minimal 5 buah lembar, yaitu sesuai dengan cara terakhir di atas. Dapatkah Anda mencari strategi yang umum untuk menyelesaikan permasalahan serupa, jika jumlah nilai uang yang dihasilkan berbeda (namun dengan pecahan-pecahan uang yang sama)? Kita bisa menganggap bahwa jumlah nilai yang diinginkan selalu merupakan kelipatan ribuan rupiah (sehingga selalu bisa didapatkan dengan menggabungkan pecahan-pecahan di atas).

Ayo Renungkan!

Setelah selesai melakukan aktivitas di atas, jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah kalian dapat memberikan sebuah contoh lain dari permasalahan optimasi yang ada di kehidupan sehari-hari?
2. Untuk contoh permasalahan yang kalian pilih sebagai jawaban di no.1, menurut kalian apakah algoritma *greedy* dapat diterapkan pada permasalahan tersebut?
3. Pada permasalahan penukaran uang di atas, apakah algoritma greedy selalu dapat digunakan untuk mencari jawaban yang paling optimal? Ambil sebuah contoh

kasus dimana pecahan yang tersedia hanya 1000, 7000 dan 10000, dan kita ingin menukarkan nilai 15.000. Berapakah jawaban yang diberikan oleh algoritma *greedy* pada kasus seperti ini? Apakah ini adalah jawaban yang optimal?

4. Untuk kasus pada pertanyaan no 3, menurut kalian, strategi seperti apakah yang kira-kira lebih tepat untuk digunakan?
5. Pelajaran paling berkesan apa yang kalian dapatkan dari latihan ini?

3. Pemrograman Dinamis

Saat menyelesaikan sebuah permasalahan optimasi (mencari nilai terbesar/terkecil), terkadang kita harus memperhitungkan beberapa kemungkinan pengambilan langkah untuk menyelesaikan permasalahan tersebut. Kemungkinan-kemungkinan tersebut mungkin memiliki akibat/konsekuensi terhadap langkah-langkah selanjutnya, sehingga pendekatan seperti teknik *greedy* mungkin tidak akan menghasilkan jawaban yang optimal. Dalam hal ini, teknik pemrograman dinamis atau *dynamic programming* (DP) mungkin akan lebih sesuai diterapkan. Teknik DP mengandung dua unsur utama, yaitu:

1. Optimasi (mencari nilai terkecil/terbesar) melalui serangkaian pilihan. Serupa dengan teknik *greedy*, kita harus menentukan rangkaian langkah apa yang akan menghasilkan nilai optimal di akhir. Namun, berbeda dengan permasalahan yang dapat diselesaikan dengan teknik *greedy*, permasalahan yang sesuai untuk teknik DP memiliki struktur sedemikian rupa sehingga pilihan langkah terbaik saat ini belum tentu merupakan pilihan terbaik secara keseluruhan, sehingga prinsip *greedy* belum tentu dapat diterapkan, dan semua kemungkinan kombinasi pilihan langkah harus diperhitungkan.

2. Nilai optimal yang diinginkan untuk permasalahan tersebut biasanya dapat dinyatakan sebagai kombinasi optimal dari sub-sub permasalahan yang sama, tetapi dengan ukuran yang lebih kecil (atau dengan kata lain, dapat dinyatakan secara rekursif). Namun, sub-sub permasalahan yang harus dipertimbangkan, biasanya memiliki *overlap* (persinggungan) sehingga dalam proses perhitungannya, diperlukan cara yang efisien untuk menghitung solusi untuk sub-sub permasalahan yang diperlukan, agar tidak terjadi perulangan/duplikasi dalam proses perhitungan. Cara yang umum digunakan adalah dengan menyimpan semua solusi dari *subproblem* yang sudah diketahui dalam sebuah tempat penyimpanan/tabel. Teknik ini biasa disebut sebagai teknik memorisasi.

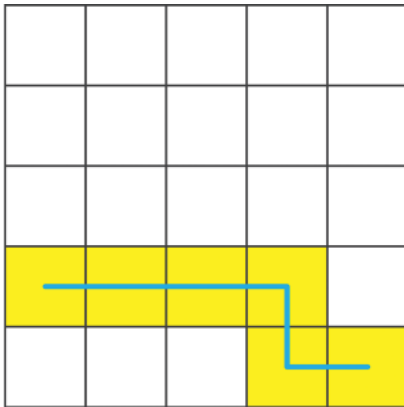
Contoh 1:

Agria ingin memanen tanaman cabai di halaman rumahnya. Tanaman tersebut ditata dalam bentuk kotak-kotak persegi seperti ilustrasi di bawah ini. Angka pada setiap kotak mewakili jumlah cabai yang ada di masing-masing tanaman.

0	1	2	3	10
10	2	10	10	5
5	3	6	5	0
0	4	0	2	3
20	5	1	0	0

Gambar 2.8 Ilustrasi Jumlah Cabai yang dipanen Agria

Agria tidak punya waktu banyak karena ia harus segera pergi ke kampus. Oleh karena itu, ia tidak bisa memetik seluruh cabai tersebut. Ia hanya bisa mulai dari kotak manapun di kolom paling kiri, dan berhenti di kotak manapun di kolom paling kanan. Agria hanya bisa bergerak ke kotak di tepat setelah kanannya

atau bawahnya. Berikut adalah salah satu dari sekian banyak kemungkinan jalur yang dapat dilalui oleh Agria untuk memetik cabai.

Gambar 2.9 Kemungkinan Jalur Memetik Cabai Agria

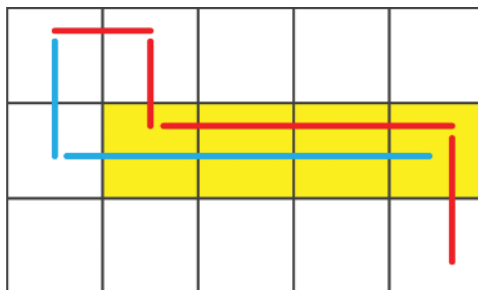
Berapakah jumlah cabai terbanyak yang bisa dikumpulkan oleh Agria?

Jawab:

Pertama, perlu dipahami bahwa penggunaan teknik *greedy* pada permasalahan ini tidak akan menghasilkan jawaban yang benar/optimal. Dapat dilihat bahwa jika kita menggunakan prinsip *greedy*, maka kita akan memilih untuk memulai dari kolom pertama baris terakhir, dengan nilai jumlah cabai terbesar, yaitu 20. Namun, jika kita memulai dari sini, maka tidak ada pilihan lain untuk langkah-langkah selanjutnya, selain bergerak terus ke kanan. Maka nilai total cabai yang akan didapatkan adalah = 20 + 5 + 1 + 0 + 0 = 26. Jelas bahwa ada pilihan-pilihan jalur lain yang akan menghasilkan total nilai cabai > 26, misalnya langkah sebagai berikut akan menghasilkan jumlah total cabai = 0 + 10 + 2 + 10 + 10 + 5 = 37.

0	1	2	3	10
10	2	10	10	5
5	3	6	5	0
0	4	0	2	3
20	5	1	0	0

Gambar 2.10 Kemungkinan Jalur Lain Memetik Cabai Agria

Metode berikutnya yang bisa kita pikirkan solusinya adalah mencoba semua kemungkinan jalur, lalu menghitung berapa nilai total cabai yang bisa didapatkan, dan terakhir mencari nilai terbesarnya. Namun dengan metode ini kita menemui kendala lainnya, yaitu akan ada terlalu banyak kemungkinan yang harus kita perhitungkan. Satu hal yang dapat kita segera pahami adalah bahwa ada banyak sekali persinggungan antara jalur-jalur yang berbeda, sedemikian rupa sehingga akan ada banyak sekali perulangan yang tidak perlu, ketika kita menghitung nilai total 0 cabai dari semua kemungkinan jalur yang ada. Sebagai contoh, kedua jalur di bawah ini (jalur biru dan jalur merah), akan melalui empat kotak yang sama dan menghitung penjumlahan dari nilai di keempat kotak yang sama tersebut (warna kuning).

Gambar 2.11 Contoh Ilustrasi Persinggungan antara dua Kemungkinan Jalur (*Overlap*)

Menggunakan prinsip DP, yang perlu kita lakukan adalah pertama-tama menyatakan solusi/penyelesaian dari permasalahan awal sebagai kombinasi dari sub-permasalahan yang lebih kecil. Dalam hal ini, kita dapat membuat argumentasi bahwa nilai jumlah cabai terbanyak yang bisa kita kumpulkan sampai dengan suatu kotak tertentu (di manapun kolomnya) tergantung dari nilai terbaik jumlah cabai sampai dengan kotak di atasnya, atau kotak di sebelah kirinya (jika ada), dan tinggal kita jumlahkan saja dengan nilai banyaknya cabai di kotak akhir tersebut. Hal ini, tentunya karena kita hanya bisa bergerak ke kanan atau ke bawah saja. Misalnya, pada Gambar 2.12 ini:

Gambar 2.12 Ilustrasi Prinsip Pemrograman Dinamis pada Persoalan Panen Cabai Agria

Nilai terbaik yang bisa kita dapatkan akan berakhir pada kotak berwarna hitam, dapat dihitung dengan cara menghitung nilai terbaik yang didapatkan sampai dengan kotak merah (misalkan nilainya = A), dan sampai dengan kotak warna biru (misalkan nilainya = B). Maka, untuk mendapatkan nilai terbaik sampai dengan kotak warna hitam, kita hanya mencari, manakah jumlah yang tertinggi antara nilai A dan B, kemudian nilai tersebut dijumlahkan dengan nilai C.

Proses di atas mengubah permasalahan ini menjadi bersifat rekursif, dimana kita bisa menggunakan hasil perhitungan pada kotak-kotak sebelumnya untuk menghitung nilai terbaik pada kotak-kotak selanjutnya (yang berada di posisi

lebih ke kanan atau ke bawah), sehingga dengan cara ini, kita bisa menghindari perulangan (duplikasi) proses perhitungan. Proses ini biasanya menggunakan sebuah tabel perhitungan yang biasa disebut sebagai tabel memoisasi (atau tabel DP). Istilah **memoisasi** berasal dari bahasa latin *memorandum* yang berarti "mengingat", yang kemudian biasa disingkat sebagai memo dalam Bahasa Inggris. Harap bedakan istilah memoisasi ini dengan memorisasi (*memorization*) yang juga memiliki arti yang serupa (proses mengingat), namun memoisasi memiliki arti yang khusus dalam dunia komputasi, yaitu: menyimpan/mengingat hasil perhitungan yang telah dilakukan sebelumnya, sehingga tidak perlu mengulang perhitungan yang sama dua kali.

Untuk soal ini, kita buat tabel memoisasi tersebut sebagai berikut:

1. Kotak paling kiri atas kita berikan nilai = nilai isi kotak tersebut (0)
2. Untuk setiap kotak lainnya, misalkan A = nilai yang sudah dihitung pada tabel memoisasi untuk kotak yang ada di atasnya (atau 0 jika kotak saat ini ada di baris teratas), dan B = nilai yang sudah dihitung pada tabel memoisasi untuk kotak yang ada di sebelah kirinya (atau 0 jika kotak saat ini ada di kolom paling kiri), serta misalkan C = nilai cabai yang ada pada kotak saat ini. Maka kita isi kotak saat ini pada tabel memoisasi dengan nilai max (A, B) + C.
3. Kita lakukan proses di atas sampai tabel memoisasi terisi penuh (sesuai ukuran tabel nilai cabai di awal). Nilai paling besar pada tabel memoisasi menunjukkan nilai total jumlah cabai terbesar yang bisa dikumpulkan.

Hasil tabel memoisasi yang sudah terisi penuh untuk soal di atas adalah sebagai berikut:

Tabel 2.2 Memoisasi Persoalan Panen Cabai Agria

0	1	3	6	16
10	12	22	32	37
15	18	28	37	37
15	22	28	39	42
35	40	41	41	42

Ayo Berlatih!

Aktivitas Berpasangan

Aktivitas SAP-K11-07-U: Bermain Angka

Deskripsi Tugas

Ani dan Budi sedang bermain dengan sebuah permainan angka: pertama Ani akan memilih sebuah angka bilangan bulat positif n. Selanjutnya, Budi harus mengubah bilangan n ini menjadi angka 1 dengan menerapkan serangkaian langkah sebagai berikut:

1. Budi boleh mengganti bilangan n dengan n - 1.
2. Jika bilangan saat ini adalah genap (habis dibagi 2), maka Budi boleh menggantinya dengan $n/2$.
3. Jika bilangan saat ini habis dibagi 3, maka Budi boleh menggantinya dengan $n/3$.

Proses ini harus dilakukan oleh Budi secara terus menerus sampai bilangan yang dimilikinya menjadi 1. Misalnya, jika Ani memilih n = 5, maka Budi dapat melakukan proses mengubah 5 menjadi 1 sebagai berikut: 5 → 4 → 2 → 1 (dalam tiga langkah). Tentukan, berapakah jumlah langkah minimum yang diperlukan, jika Ani memilih n = 25?

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Menurut kalian, apakah permasalahan penukaran uang pada aktivitas SAP-K11-06-U: Menukarkan Uang untuk topik greedy sebelumnya (yaitu pecahan yang tersedia hanya 1000, 7000 dan 10000, dan kita ingin menukarkan nilai 15.000) dapat diselesaikan dengan teknik pemrograman dinamis? Bagaimana caranya?
2. Apakah ada contoh permasalahan yang lain dalam kehidupan sehari-hari yang dapat diselesaikan dengan teknik pemrograman dinamis (atau setidaknya, dimana teknik memoisasi dapat diterapkan)?
3. Menurut Anda, apakah yang menjadi perbedaan penting antara algoritma greedy dan pemrograman dinamis? Adakah kekurangan dan kelebihan masing-masing?
4. Manakah pelajaran yang paling berkesan dari topik ini?

C. Algoritma dan Pemrograman

Pada jenjang sebelumnya dalam materi Informatika Kelas X, kalian telah mempelajari mengenai beberapa bahasa pemrograman, salah satunya adalah bahasa C yang telah dipelajari pada elemen Algoritma dan Pemrograman. Kemampuan bahasa pemrograman terutama bahasa C akan kalian gunakan dalam topik ini. Selanjutnya untuk lebih meningkatkan pemahaman kalian dalam elemen Algoritma dan Pemrograman, kalian akan mempelajari konsep larik serta karakter dan string. Kedua konsep ini sangat penting dalam membuat program dan akan kalian gunakan untuk menyelesaikan berbagai permasalahan yang diberikan pada bagian D dalam bab ini.

1. Larik (*Array*)

Saat ini mungkin kalian memiliki pertanyaan seperti "mengapa contoh dan permasalahan yang diberikan selama ini adalah hal yang dapat diselesaikan oleh manusia secara manual atau dengan kalkulator?" Jawabannya adalah karena permasalahan/*problem* tersebut diberikan untuk mendukung proses kalian menguasai kompetensi dalam algoritma dan pemrograman.

Pada praktiknya, program digunakan untuk mengolah data yang berukuran besar dan membutuhkan waktu yang sangat lama jika dikerjakan manual oleh manusia. Misalnya kalian perlu menghitung statistika deskriptif (seperti rata-rata, nilai minimal, nilai maksimal, standar deviasi, dan sebagainya) dari data seluruh penduduk Indonesia. Walaupun masalahnya sederhana, namun dikarenakan jumlah data yang diolah sangat banyak dan berukuran besar mengakibatkan waktu pengerjaan pun menjadi sangat lama bagi manusia untuk menyelesaikannya. Bahkan ada kemungkinan data berukuran besar tersebut tidak dapat diolah menggunakan aplikasi pengolah data (*spreadsheet*) yang tidak dirancang untuk mengolah data sebesar itu. Untuk solusinya kalian dapat menggunakan program khusus untuk mengolah data berukuran besar atau membuat program sendiri yang mampu menyimpan dan mengolah data berukuran besar.

Kita pun sampai pada pertanyaan besar: "bagaimana caranya membuat program yang mampu menyimpan dan mengolah data berukuran besar?". Sebelumnya kalian telah mengenal konsep variabel yang mampu menyimpan satu buah nilai dengan tipe data tertentu (variabel tunggal). Permasalahan akan muncul ketika program kita harus mengolah sebanyak satu juta data, apakah kita harus membuat satu juta variabel? Bukankah hal tersebut sangat sulit dipraktikkan dalam kode program yang kita tulis? Untuk mengatasi hal tersebut, bahasa pemrograman

memiliki suatu alat untuk menyimpan himpunan data ke dalam satu nama variabel yang diberikan indeks. Salah satunya disebut sebagai **larik** atau ***array***.

Contoh di dunia nyata yang merepresentasikan larik adalah seperti loker yang diberikan nomor (Gambar 2.13). Ketika kalian menyimpan barang di loker tersebut, kalian akan mengingat nomor loker tersebut. Kalian juga dapat menyimpan barang di loker dengan nomor yang berbeda. Pada analogi tersebut, nomor pada loker adalah indeks yang kita gunakan untuk mengenali tempat kita menyimpan barang kita tadi.

Gambar 2.13 Loker sebagai Ilustrasi Larik dalam Pemrograman

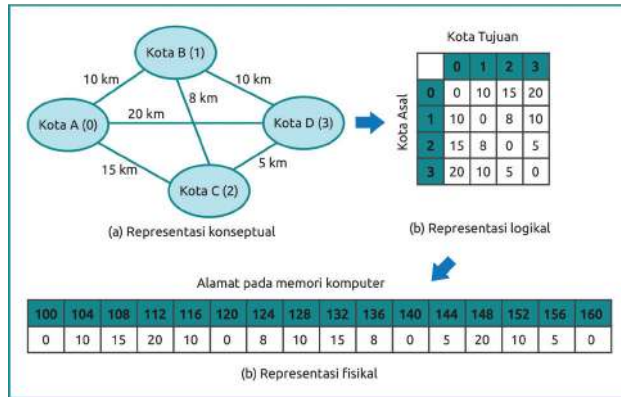
Menggunakan larik pada program mirip dengan membuat variabel tunggal. Perbedaannya adalah kalian perlu memberikan informasi ukuran dari larik yang akan dibuat dan kalian perlu mengaksesnya dengan menggunakan indeks. Ada banyak cara untuk mengimplementasikan konsep larik ke dalam kode program, salah satunya menggunakan *vector*. *Vector* ini merupakan pustaka bahasa pemrograman C++ yang juga dapat digunakan pada bahasa C. *Vector* dapat dibuat untuk menampung himpunan data yang ukurannya dinamis bergantung pada jumlah (atau ukuran) data yang disimpan di dalamnya.

Contoh kode program di bawah ini mengilustrasikan cara mendeklarasikan, menginisialisasi, menyimpan nilai ke, dan mengakses nilai pada suatu *vector*. Berbeda dengan buku Kelas X, program ini ditulis menggunakan bahasa pemrograman C++. Oleh karena itu, kalian perlu menyimpan kode program dengan ekstensi berkas .cpp dan mengkompilasinya dengan kompilator bahasa C++, misalnya GCC.

```
1:   /* Demonstrasi larik satu dimensi dengan vektor */
2:
3:   #include <stdio.h>
4:   #include <vector>
5:   using namespace std;
6:
7:   void cetakLarik(int ukuran, vector<int> larik){
8:       int i;
9:       for(i = 0; i < ukuran; i++)
10:          printf("%d\n", larik[i]);
11:  }
12:
13:  int main(){
14:      vector<int> bilangan;
15:      int i;
16:      for(i = 0; i < 10; i++)
17:          bilangan.push_back(i);
18:      cetakLarik(10, bilangan);
19:  }
```

Kalian juga dapat membuat larik yang berukuran dua dimensi, yang memiliki ukuran berupa baris dan kolom (seperti pada matriks di matematika). Pada kasus seperti apa larik dua dimensi digunakan? Misalnya kalian ingin menghitung jarak antara dua kota yang dihubungkan dengan jalan dua arah. Kalian dapat menggunakan matriks dua dimensi untuk merepresentasikan jarak antara dua kota seperti pada Gambar 2.14 berikut.

	0	1	2	3
0	0	10	15	20
1	10	0	8	10
2	15	8	0	5
3	20	10	5	0

100	104	108	112	116	120	124	128	132	136	140	144	148	152	156	160
0	10	15	20	10	0	8	10	15	8	0	5	20	10	5	0

Gambar 2.14 Ilustrasi representasi panjang jalan penghubung antar empat kota (A-D) dalam bentuk matriks dua dimensi.

Gambar tersebut memperlihatkan proses berpikir komputasional yaitu abstraksi. Suatu hubungan antara kota di dunia nyata (kontekstual) dibuat dalam bentuk yang lebih sederhana dalam bentuk grafik yang menampilkan informasi yang relevan (konseptual), yaitu hubungan antar kota dan jaraknya. Setelah itu, bentuk tadi diubah ke dalam bentuk yang dapat digunakan dalam menyusun algoritma komputer (logikal) dalam bentuk matriks dua dimensi. Di balik program yang berjalan di komputer kalian, program akan menyimpan matriks tersebut di dalam memori komputer (fisikal) yang memiliki bentuk seperti matriks satu dimensi.

Implementasi larik bisa berbeda-beda di bahasa C atau bahasa lain, yang penting adalah implementasi dari kode program tersebut memenuhi perilaku larik, yaitu adanya himpunan data yang disimpan dan dapat diakses dengan menggunakan suatu indeks. Misalnya, kalian kemudian dapat membuat kode program yang akan menerima isi matriks jarak di atas dan menyimpannya dalam sebuah vector dua dimensi. Setelah itu, program akan dapat menjawab jarak antara dua kota yang ditanyakan oleh pengguna. Perhatikanlah potongan kode program berikut. Setelah itu, perkuat pemahaman kalian dengan latihan yang diberikan.

```
1:   /* Program demonstrasi penggunaan larik dua dimensi */
2:
3:   #include <stdio.h>
4:   #include <vector>
5:   #define JUMLAH_KOTA 4
6:   using namespace std;
7:
8:   int main(){
9:       vector<vector<int>> jarak;
10:      int i, j;
11:      int asal, tujuan;
12:
13:      for(i = 0; i < JUMLAH_KOTA; i++){
14:          vector<int> vtr;
15:          for(j = 0; j < JUMLAH_KOTA; j++){
16:              int x;
17:              scanf("%d",&x);
18:              vtr.push_back(x);
19:          }
20:          jarak.push_back(vtr);
21:          vtr.clear();
22:      }
23:
24:      scanf("%d %d", &asal, &tujuan);
25:      printf("Jarak Kota-%d Kota -%d = ", asal, tujuan);
26:      printf("%d km\n", jarak[asal][tujuan]);
27:  }
```

Ayo Berlatih!

Aktivitas Individu

Aktivitas SAP-K11-08: Latihan Larik

1. Rancang dan buatlah sebuah program yang membaca N buah bilangan dan mencetaknya secara terbalik. Misalnya jika diberikan masukan 1 2 3 4 5 6, program akan mencetak 6 5 4 3 2 1.

2. Rancang dan buatlah sebuah program yang membaca sebuah matriks berukuran $N \times M$ dan mencetak hasil *transpose* matriksnya.

3. Modifikasilah kode program untuk menghitung jarak dua kota yang menerima suatu rute. Misalnya, jika rute yang dimasukkan adalah Kota A - Kota B - Kota C, maka nilai total jarak yang dikeluarkan adalah 10 + 8 = 18 km.

4. Tantangan larik: perhatikan bahwa pada Gambar 2.9 terdapat data yang berganda. Hal ini dikarenakan matriks tersebut menyimpan informasi jarak antara 2 kota (misal A dan B) sebagai jarak dari A-B dan B-A. Dapatkan kalian menemukan representasi yang lebih baik daripada contoh tersebut sehingga tidak ada duplikasi informasi dalam penyimpanan data jarak kota?

Ingin Tahu Lebih?

1. Masih banyak detail dan fitur dari *vector* yang tidak dapat dijelaskan lengkap di buku ini. Kalian dapat membacanya lebih lanjut di dokumentasi *vector* di bahasa C++ berikut: en.cppreference.com/w/cpp/container/vector
2. Selain menggunakan vector, larik pada bahasa C dapat diimplementasikan menggunakan *array* bawaan dari bahasa pemrograman C. Dokumentasi larik dalam bahasa C dapat diakses di en.cppreference.com/w/c/language/array.
3. Larik adalah cara yang paling sederhana untuk menyimpan himpunan data. Masih ada banyak cara lain untuk menyimpan dan mengolah data, yang kita sebut sebagai struktur data. Kalian dapat melihat ilustrasi visual dari bagaimana beberapa struktur data bekerja di: visualgo.net/id.

Praktik Baik Pemrograman

Saat memprogram, kalian akan menggunakan suatu bahasa pemrograman, pustaka, atau perangkat tertentu. Biasakanlah untuk selalu melihat dokumentasi yang tersedia dari teknologi yang kalian gunakan. Dokumentasi ini disediakan oleh pengembang teknologi untuk memberikan informasi tentang teknologi yang mereka sediakan. Pada bahasa pemrograman misalnya kalian dapat melihat penjelasan sintaks dan fitur-fitur terkait bahasa pemrograman yang kalian gunakan. Ada banyak fitur menarik yang dapat kalian eksplorasi dan manfaatkan di program yang kalian buat.

2. Karakter dan *String*

Selain angka, masukan dari program dapat berupa karakter atau rangkaian karakter seperti kata. Hal ini telah lazim kalian temui, misalnya ketika mengetikkan kata kunci untuk melakukan pencarian di mesin pencari atau ketika kalian memasukkan kata sandi saat *login*. Oleh karena itu, program dilengkapi dengan kemampuan untuk membaca, menyimpan, mengolah, dan mencetak rangkaian karakter tersebut. Rangkaian karakter tersebut dalam pemrograman disebut sebagai ***string***.

Definisi *string* dalam pemrograman adalah rangkaian karakter. Karakter sendiri merupakan suatu data berupa huruf, angka, simbol, dan karakter lain yang mengikuti suatu standar tertentu seperti "**American Standard Code for Information Interchange**" (ASCII) atau Unicode. Karakter pada pemrograman pada umumnya diimplementasikan dalam program menggunakan tipe data *char*. Di sisi lain, ada beberapa cara yang lazim digunakan untuk mengimplementasikan string dalam pemrograman.

Pada bahasa C, *string* diimplementasikan sebagai larik karakter yang diakhiri oleh karakter '\0'. Dengan kata lain, kalian dapat membuat dan memproses suatu *string* seperti halnya kalian mengolah larik. Bahasa pemrograman lain seperti C++ atau Java memilih sebuah tipe data string sendiri yang menyembunyikan beberapa detail terkait pengelolaan data *string* yang dilakukan oleh program.

Karena proses pada string berbeda dengan proses pada bilangan, untuk itu bahasa pemrograman telah dilengkapi dengan fungsi-fungsi untuk mengolah karakter dan *string*. Misalnya untuk melakukan konversi dari huruf kapital ke non kapital, penggabungan *string*, pencarian *substring*, dan berbagai fungsi lainnya. Pada bahasa C, kalian dapat akses fungsi-fungsi tersebut pada pustaka <string.h>. Pada contoh

berikut, diberikan dua buah kode program C++ untuk mengolah karakter dan string.

Program pertama akan membaca sebuah *string*, mengubahnya ke dalam huruf non kapital, lalu mencetaknya:

```
1:   /* Demonstrasi karakter dan string */
2:
3:   #include <stdio.h>
4:   #include <iostream>
5:   #include <string.h>
6:   using namespace std;
7:
8:   int main(){
9:       string str;
10:      cin >> str;
11:      for(int i = 0; i < str.length(); i++)
12:          str[i] = tolower(str[i]);
13:
14:      cout << str << '\n';
15:  }
```

Program kedua akan membaca banyak *string* hingga membaca string “STOP”. Setiap *string* yang dibaca akan diubah ke huruf non kapital dan dicetak.

```
1:   /* Demonstrasi karakter dan string */
2:
3:   #include <stdio.h>
4:   #include <iostream>
5:   #include <string.h>
6:   using namespace std;
7:
8:   string lowerCase(string str){
9:       for(int i = 0; i < str.length(); i++)
10:          str[i] = towlower(str[i]);
11:      return str;
12:  }
13:
14:  int main(){
15:      string str;
16:      cin >> str;
17:      while(str != "STOP"){
18:          cout << lowerCase(str) << '\n';
19:          cin >> str;
20:      }
21:  }
```

Telusurilah kedua program tersebut untuk memahami seperti apa mereka bekerja. Kemudian, salin dan jalankan kedua program tersebut menggunakan kompilator bahasa C++. Jika diperlukan, pelajarilah dokumentasi pustaka string.h di C dan tipe data *string* pada C++ seperti yang dicantumkan pada bagian Ingin Tahu Lebih.

Ayo Berlatih!

Aktivitas Individu

Aktivitas SAP-K11-09: Latihan Karakter dan String

1. Rancang dan buatlah sebuah program yang dapat membaca sebuah string dan mencetaknya secara terbalik.
2. Rancang dan buatlah sebuah program yang membaca sebuah kata sandi dan mencetak jumlah karakter yang berupa angka, huruf, dan simbol. Bedakan huruf kapital dan non kapital.

Ingin Tahu Lebih?

1. Referensi terhadap fungsi-fungsi yang terdapat pada pustaka *string*.h dapat diakses di en.cppreference.com/w/cpp/header/cstring.
2. Referensi terhadap tipe data *string* pada bahasa pemrograman C++ dapat diakses di en.cppreference.com/w/cpp/string/basic_string.

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Seberapa paham kalian dengan konsep larik pada pemrograman?
2. Seberapa paham kalian dengan konsep karakter dan string pada pemrograman?
3. Seperti apa pengalaman yang kalian rasakan saat membaca dokumentasi bahasa pemrograman yang disediakan?
4. Pelajaran paling berkesan yang kalian dapatkan dari aktivitas latihan ini?

D. Latihan Strategi Algoritmik dan Pemrograman Lintas Bidang

Pada bagian ini kalian akan membuat berbagai program berdasarkan permasalahan yang tersedia, tiap permasalahan memiliki sub permasalahannya tersendiri yang tingkat kesulitannya meningkat. Kalian akan membuat program dimulai dari perancangan, yaitu merancang algoritma untuk menyelesaikan permasalahan tersebut. Selanjutnya algoritma tersebut kalian terjemahkan ke dalam bahasa pemrograman yang kalian kuasai, misalnya bahasa C, Python,dsb. Permasalahan tersebut akan meningkatkan kemampuan programming kalian dengan mempelajari bagian ini dengan menyelesaikan berbagai sub permasalahan yang tersedia.

1. *Problem* Simulasi Burung

Pada bagian ini, kalian akan membuat program untuk mensimulasikan gerak burung yang diluncurkan dengan menggunakan alat ketapel. Secara prinsip, gerakan burung yang diluncurkan dengan menggunakan ketapel menggunakan prinsip gerak lurus berubah beraturan (GLBB). Terdapat komponen sudut, gravitasi serta kecepatan dan waktu yang menjadi penentu jauhnya burung tersebut dapat meluncur dengan menggunakan ketapel. Konsep ini menggunakan kaidah gerak parabola yang telah kalian pelajari pada Mata

Pelajaran Fisika. *Problem* akan dibagi menjadi beberapa *subproblem* dengan tingkat kesulitan yang meningkat. Ikutilah petunjuk guru kalian dalam memilih tingkat kesulitan *subproblem* yang akan kalian kerjakan. Apabila kalian berhasil mengerjakan *subproblem* tersebut, kalian dapat menantang diri kalian untuk mengerjakan subproblem yang lebih sulit.

Setelah ditentukan tingkat kesulitan subproblem yang akan dikerjakan, kalian dapat mulai mengerjakan aktivitas Ayo Merancang Program: Merancang Algoritma Simulasi Burung dan Ayo Buat Program: Membuat Program Simulasi Burung berdasarkan deskripsi permasalahan yang sesuai.

Gambar 2.15
Ilustrasi Boro yang sedang tidur

Problem: Program Simulasi Burung

Boro adalah seekor burung yang terjatuh dari sarangnya di sebuah pohon saat sedang tidur. Saat terjatuh, sayap Boro menghantam tanah dan ia sangat merasa kesakitan. Boro harus kembali ke sarangnya, namun karena sayapnya terluka, ia tidak dapat terbang sebagaimana mestinya. Tak jauh dari tempat Boro jatuh, terdapat ketapel raksasa yang biasa digunakan oleh pemilik lahan untuk kegiatan sirkus. Ketapel tersebut biasa digunakan untuk kegiatan menembak dengan menggunakan buah semangka.

Kebetulan saat itu, pemilik lahan sedang latihan sirkus menembak semangka dengan menggunakan ketapel. Boro meminta tolong kepada pemilik lahan, apakah ia dapat ikut di atas semangka tersebut agar dapat kembali ke rumahnya di atas pohon. Sang pemilik lahan setuju akan membantu dan menanyakan kepada Boro tentang lokasi pohon tempat tinggalnya.

Subproblem 1: Menghitung Jarak Horizontal Terjauh

Deskripsi:

Pada *subproblem* ini, kalian akan diminta untuk menghitung jarak horizontal terjauh yang dapat ditempuh oleh Boro apabila Boro ikut berdiri di atas semangka yang akan diluncurkan oleh pemilik lahan.

Format Masukan:

Baris pertama adalah sebuah bilangan bulat S yang menggambarkan sudut peluncuran. Nilai S ini bernilai 0–90. Baris kedua adalah V yang merupakan kecepatan awal Boro saat meluncur dengan menggunakan ketapel. Asumsikan bahwa nilai gravitasi adalah 10

Format Keluaran:

Keluaran berupa bilangan yang menunjukkan jarak terjauh Boro mendarat di tanah.

Tabel 2.3 Format Keluaran Program Simulasi Burung *Subproblem* 1

Contoh Masukan	Contoh Keluaran
37 10	9.6

Subproblem 2: Menghitung Waktu

Deskripsi:

Pada subproblem ini, kalian akan diminta untuk menghitung waktu yang diperlukan Boro untuk mencapai jarak horizontal terjauh apabila Boro ikut berdiri di atas semangka yang akan diluncurkan oleh pemilik lahan.

Format Masukan:

Baris pertama adalah sebuah bilangan bulat S yang menggambarkan sudut peluncuran. Nilai S ini bernilai 0–90. Baris kedua adalah V yang merupakan kecepatan awal Boro saat meluncur dengan menggunakan ketapel. Asumsikan bahwa nilai gravitasi adalah 10

Format Keluaran:

Keluaran berupa bilangan yang menunjukkan waktu yang ditempuh Boro untuk mencapai jarak terjauh dengan format 3 angka di belakang koma.

Tabel 2.4 Format Keluaran Program Simulasi Burung *Subproblem* 2

Contoh Masukan	Contoh Keluaran
37 100	12.036

Subproblem 3: Prediksi Ketinggian Dicapai Boro

Deskripsi:

Pada *subproblem* ini, kalian akan diminta untuk memberikan prediksi apakah Boro dapat mencapai ketinggian lebih tinggi daripada tinggi pohon tempat ia bersarang, apabila Boro ikut berdiri di atas semangka yang akan diluncurkan oleh pemilik lahan.

Format Masukan:

Baris pertama adalah sebuah bilangan bulat S yang menggambarkan sudut peluncuran. Nilai S ini bernilai 0–90. Baris kedua adalah V yang merupakan kecepatan awal Boro saat meluncur dengan menggunakan ketapel. Baris ketiga adalah T yang merupakan tinggi pohon tempat Boro Bersarang. Asumsikan bahwa nilai gravitasi adalah 10.

Format Keluaran:

Keluaran berupa bilangan yang menunjukkan status ketinggian Boro dibandingkan dengan tinggi pohon tempat sarang Boro. 1 apabila Boro dapat mencapai ketinggian sama dengan tinggi pohon tempat sarangnya berada atau lebih, 0 apabila Boro tidak mampu mencapai ketinggian yang sama dengan pohon tersebut. Serta ketinggian maksimum yang dapat diperoleh oleh Boro saat meluncur dengan menggunakan ketapel.

Tabel 2.5 Format Keluaran Program Simulasi Burung *Subproblem* 3

Contoh Masukan	Contoh Keluaran
37 100 100	Status : 1 Ketinggian : 181.09
37 100 200	Status : 0 Ketinggian : 181.09

Subproblem 4: Prediksi Ketinggian Dicapai Boro dan Teman-temannya

Deskripsi:

Pada subproblem ini, kalian akan diminta untuk memberikan prediksi apakah Boro dan teman-temannya dapat mencapai ketinggian lebih tinggi daripada tinggi pohon tempat ia bersarang. Pada *subproblem* ini, burung yang akan menumpang semangga tidak hanya Boro, namun terdapat burung lain yang akan menumpang semangka tersebut secara bergantian.

Format Masukan:

Baris pertama adalah sebuah bilangan bulat N yang menggambarkan jumlah burung yang akan ikut peluncuran. Baris kedua adalah S yang menggambarkan sudut peluncuran. Nilai S ini bernilai 0-90. Baris ketiga adalah V yang merupakan kecepatan awal Burung saat meluncur dengan menggunakan ketapel. Baris keempat adalah T yang merupakan tinggi pohon tempat burung bersarang. Asumsikan bahwa nilai gravitasi adalah 10.

Format Keluaran:

Keluaran berupa urutan burung lalu bilangan yang menunjukkan status ketinggian burung dibandingkan dengan tinggi pohon tempat sarang burung tersebut. 1 apabila burung dapat mencapai ketinggian sama dengan

tinggi pohon tempat sarangnya berada atau lebih, 0 apabila burung tidak mampu mencapai ketinggian yang sama dengan pohon tersebut. Serta ketinggian maksimum yang dapat diperoleh oleh burung saat meluncur dengan menggunakan ketapel.

Tabel 2.6 Format Keluaran Program Simulasi Burung *Subproblem* 4

Contoh Masukan	Contoh Keluaran
1 37	Status Burung 1 : 1 Ketinggian: 181.09
100 100	Status Burung 1 : 1 Ketinggian: 181.09
2 37 100 100 37 100 200	Status Burung 2 : 0 Ketinggian: 181.09

Ayo Merancang Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-10-U: Merancang Algoritma Simulasi Burung

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang simulasi burung yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam Buku Kerja kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah

itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Ayo Buat Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-11: Membuat Program Simulasi Burung

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apa yang kalian rasakan saat membuat suatu algoritma dan program dari suatu permasalahan di mata pelajaran lain yang telah kalian ikuti?
2. Apakah program yang kalian buat dapat membantu kalian atau orang lain?
3. Apakah solusi yang telah kalian buat dapat dimanfaatkan untuk menyelesaikan masalah lain yang sejenis?
4. Adakah pengembangan lebih lanjut (enhancing) yang terpikir oleh kalian agar program menjadi lebih bermanfaat?
5. Apa yang kalian rasakan saat memeriksa solusi algoritma dan program teman kalian?
6. Apa yang kalian rasakan saat solusi algoritma dan program kalian diperiksa oleh teman kalian?
7. Pelajaran paling berkesan apa yang kalian dapatkan dari latihan ini?

2. *Problem* Pengelolaan Bank Darah

Pada bagian ini, kalian akan menyelesaikan suatu permasalahan yang terkait dengan golongan darah. Informasi mengenai golongan darah sangat berguna di bidang kesehatan, bahkan dapat menyelamatkan nyawa seseorang. Dalam problem ini, kalian akan membantu sebuah rumah sakit untuk memastikan apakah stok darah yang mereka miliki cukup.

Problem: Pengelolaan Bank Darah

Suatu Bank Darah di Rumah Sakit mengelola stok darah yang mereka miliki. Persediaan darah yang mereka miliki dikelompokkan berdasarkan golongan darah standar A, AB, B, dan O seperti yang selama ini kita kenal. Kondisi stok darah yang aman adalah tersedianya cadangan yang cukup untuk setiap golongan darah. Apabila terdapat golongan darah yang stoknya menipis atau habis, maka

pihak Bank Darah akan melakukan kegiatan bakti donor darah di suatu desa. Masalah ini di dunia nyata lebih kompleks karena mempertimbangkan rhesus darah dan faktor lainnya, namun kita sederhanakan dengan hanya mempertimbangkan golongan darah pendonor.

Agar kegiatan bakti donor darah dapat berjalan secara efektif dan tepat sasaran, pihak Bank Darah mencatat data para Donor Darah Sukarela (DDS) yang tersebar di berbagai daerah. Informasi mengenai DDS yang dicatat oleh pihak Bank Darah adalah informasi kode desa, golongan darah, dan volume darah yang dapat didonorkan oleh DDS. Contoh data tersebut adalah sebagai berikut:

Tabel 2.7 Contoh Data DDS Program Pengelolaan Bank Darah

Kode DDS	1	2	3	4	5	6	7	8
Kode Desa	3	1	3	1	2	3	1	2
Golongan Darah	A	B	A	AB	B	O	O	A
Volume Darah yang Dapat Didonorkan (ml)	150	250	300	450	200	350	500	500

Karena semakin lama, data DDS yang harus dikelola oleh Bank Darah tersebut semakin banyak, bahkan bisa jutaan orang, mereka meminta kalian untuk membangun sebuah program yang dapat menyimpan dan mengolah data DDS tersebut. Hasil pengolahan data tersebut akan digunakan oleh Bank Darah untuk menentukan pelaksanaan kegiatan Bakti Donor Darah yang lebih efektif dan tepat sasaran. Program tersebut harus bisa menentukan desa-desa yang harus dikunjungi untuk diadakan kegiatan bakti donor darah. Agar kalian merasakan proses pemrograman yang iteratif, kalian akan membuat program tersebut melalui empat tahap berikut:

***Subproblem* 1**

Program mampu menghitung jumlah pendonor darah di setiap desa.

↓

***Subproblem* 2**

Program mampu menghitung jumlah pendonor darah dan volume darah yang dapat didonorkan di masing-masing desa.

↓

***Subproblem* 3**

Program mampu menentukan desa-desa yang perlu diadakan kegiatan bakti donor darah sehingga kebutuhan darah terpenuhi.

↓

***Subproblem* 4**

Seperti subproblem 3, tapi jumlah desa yang diolah mencakup seluruh Indonesia.

Gambar 2.16 Empat Tahap Pembuatan Program Pengelolaan Bank Darah

Catatan: Untuk menghindari kesulitan dalam entri data golongan darah AB, *problem* ini akan memberikan kode golongan darah A = 1, B = 2, AB = 3, dan O = 4. Kalian boleh saja mengkode dengan kode lainnya dengan syarat harus konsisten.

Subproblem 1: Menghitung Jumlah Pendonor Darah

Deskripsi:

Pada *subproblem* ini, kalian akan diminta untuk menghitung banyaknya pendonor darah di setiap desa berdasarkan kode desa yang diberikan.

Format Masukan:

Baris pertama adalah sebuah bilangan bulat N yang merupakan jumlah DDS. Nilai N ini paling banyak berjumlah 1000 orang. Baris kedua adalah M yang merupakan jumlah desa yang paling banyak berjumlah

10 desa. N baris berikutnya masing-masing akan terdiri atas tiga buah informasi, yaitu kode desa, golongan darah, dan volume darah yang dapat didonorkan. Kode desa merupakan sebuah bilangan bulat dari 1 sampai M, sedangkan volume darah yang dapat didonorkan adalah sebuah bilangan bulat dari 0 hingga 500 ml.

Format Keluaran:

Keluaran berupa banyaknya pendonor yang ada di masing-masing desa. Jumlah pendonor dicetak dengan format Desa M: Banyaknya Pendonor.

Tabel 2.8 Format Keluaran Program Pengelolaan Bank Darah Subproblem 1

Contoh Masukan	Contoh Keluaran
8 3 3 1 150 1 2 250 3 1 300 1 3 450 2 2 200 3 4 350 1 4 500 2 1 500	Desa 1: 3 Desa 2: 2 Desa 3: 3

Subproblem 2: Menghitung Jumlah Pendonor dan Total Volume Darah

Deskripsi:

Pada *subproblem* ini, kalian akan diminta untuk menghitung jumlah pendonor darah dan total volume darah yang dapat didonorkan di setiap desa berdasarkan kode desa dan golongan darah.

Format Masukan:

Sama dengan *subproblem* 1.

Format Keluaran:

Keluaran berupa jumlah pendonor dan total volume darah yang ada di masing-masing desa berdasarkan golongan darahnya. Ikuti format yang ada di contoh keluaran berikut:

Tabel 2.9 Format Keluaran Program Pengelolaan Bank Darah Subproblem 2

Contoh Masukan	Contoh Keluaran
8	Desa 1:
3	A: 0 0
3 1 150	B: 1 250
1 2 250	AB: 1 450
3 1 300	O: 1 500
1 3 450	Desa 2:
2 2 200	A: 1 500
3 4 350	B: 1 200
1 4 500	AB: 0 0
2 1 500	O: 0 0
	Desa 3:
	A: 2 450
	B: 0 0
	AB: 0 0
	O: 1 350

Subproblem 3: Memilih Tempat Pelaksanaan Donor Darah

Deskripsi:

Pada *subproblem* ini, kalian akan diminta membantu Bank Darah dalam memilih tempat pelaksanaan Bakti Donor Darah ketika stok suatu golongan darah habis. Karena pelaksanaan bakti donor darah membutuhkan biaya, kalian harus meminimumkan jumlah desa dimana bakti donor darah akan dilaksanakan, dengan tetap mencapai jumlah volume darah yang dibutuhkan.

Format Masukan:

Sama dengan *subproblem* 1. Setelah seluruh data DDS dimasukkan, baris berikutnya terdiri atas dua bilangan yang merupakan kode golongan darah yang habis, dan volume darah yang dibutuhkan (dalam ml). Nilai volume darah ini paling banyak bernilai satu juta mililiter.

Format Keluaran:

Apabila kebutuhan darah dapat dipenuhi, cetak daftar kode-kode desa yang akan dilaksanakan bakti golongan darah beserta jumlah darah yang dibutuhkan, dan satu kalimat "Kebutuhan darah dipenuhi dengan surplus X ml". Sebaliknya, apabila kebutuhan darah tidak dapat dipenuhi, cetak "Kebutuhan darah masih kurang X ml" setelah seluruh keluaran dicetak.

Misalnya pada contoh kasus uji di *subproblem* 2, jika ditambahkan dengan 2 300 (artinya, ada kebutuhan 300ml golongan darah B) maka program haruslah menghasilkan keluaran:

```
Desa 1  : 250 ml
Desa 2  : 200 ml
Kebutuhan darah dipenuhi dengan surplus 150 ml
```

Karena dengan mengadakan bakti donor darah di kedua desa (Desa 1 dan Desa 2), kita dapat mengumpulkan total 450ml darah golongan B, sehingga menyisakan kelebihan 150ml. Tidak ada cara lain yang dapat dilakukan untuk memenuhi kebutuhan darah golongan B dengan mengadakan bakti donor darah di satu desa saja.

Subproblem 4: Tingkatkan Penyimpanan dan Kecepatan Olah Data DDS

Deskripsi:

Subproblem 4 sama dengan Subproblem 3. Hanya saja yang berbeda adalah ukuran problem yang diberikan. Program kalian sekarang harus bisa menyimpan dan

mengolah data para Donor Darah Sukarela dari seluruh Indonesia, sehingga batasan masukan naik menjadi:

Jumlah DDS : maksimal 30 juta orang
Jumlah Desa : maksimal 100000 desa

Selain itu, karena pengambilan keputusan harus diambil secara cepat, program kalian harus dapat berjalan dalam waktu paling lama 1 detik. Buatlah program yang dapat melakukan hal tersebut.

Ayo Merancang Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-12-U: Merancang Algoritma Pengelolaan Donor Darah

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang pengelolaan donor darah yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam Buku Kerja kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Ayo Buat Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-13: Membuat Program Pengelolaan Donor Darah

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Ini Berpikir Komputasional!

Kegiatan yang telah kalian kerjakan di atas adalah menerapkan pemrograman untuk menyelesaikan permasalahan yang ada di dunia nyata. Terlihat bahwa informatika dapat digunakan untuk mengambil keputusan yang lebih baik berdasarkan data akurat yang diolah dengan cepat. Tentunya, problem yang diberikan di atas adalah penyederhanaan dari problem yang ada di lapangan, yang memiliki kompleksitas yang lebih tinggi.

Pada saat membaca deskripsi subproblem keempat, kalian mungkin kaget melihat jumlah data yang sangat besar. Program yang telah kalian buat untuk mengerjakan Subproblem 3 belum tentu dapat mengerjakan Subproblem 4 dalam waktu maksimal 1 detik yang diberikan. Yang berbeda dari Subproblem 3 dengan Subproblem 4 adalah ukuran dari problem yang lebih besar. Di sinilah salah satu tantangan dari big data, yaitu bagaimana kita dapat merancang algoritma yang paling efisien sehingga data yang besar tadi tetap dapat diolah dalam waktu yang singkat.

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apa yang kalian rasakan saat membuat algoritma dan program untuk membantu suatu lembaga?
2. Apakah program yang kalian buat dapat membantu kalian atau orang lain?
3. Apakah solusi yang telah kalian buat dapat dimanfaatkan untuk menyelesaikan masalah lain yang sejenis?
4. Adakah pengembangan lebih lanjut (enhancing) yang terpikir oleh kalian agar program menjadi lebih bermanfaat?
5. Apa yang kalian rasakan saat memeriksa solusi algoritma dan program teman kalian?
6. Apa yang kalian rasakan saat solusi algoritma dan program kalian diperiksa oleh teman kalian?
7. Prinsip Berpikir Komputasional apa saja yang kalian terapkan saat menyelesaikan aktivitas di atas?
8. Pelajaran paling berkesan apa yang kalian dapatkan dari latihan ini?

3. *Problem* Persilangan Tanaman

Pada bagian ini, kalian akan membuat program untuk mensimulasikan persilangan tanaman menggunakan hukum Mendel yang telah kalian pelajari pada Mata Pelajaran Biologi. *Problem* akan dibagi menjadi beberapa *subproblem* dengan tingkat kesulitan yang meningkat. Ikutilah petunjuk guru kalian dalam memilih tingkat kesulitan *subproblem* yang akan kalian kerjakan. Apabila kalian berhasil mengerjakan *subproblem*

tersebut, kalian dapat menantang diri kalian untuk mengerjakan *subproblem* yang lebih sulit.

Setelah ditentukan tingkat kesulitan *subproblem* yang diberikan, kalian dapat mengerjakan aktivitas Ayo Merancang Program: Merancang Algoritma Persilangan Tanaman dan Ayo Buat Program: Membuat Program Persilangan Tanaman berdasarkan deskripsi permasalahan yang sesuai.

Problem: Persilangan Benih Padi Unggulan

Tania, seorang peneliti di bidang pemuliaan tanaman, sedang melakukan persilangan tanaman padi untuk menghasilkan benih unggulan. Sifat tanaman padi yang akan ia perhatikan adalah tinggi tanaman, jumlah anakan, jumlah bulir, dan ketahanan terhadap penyakit. Seluruh sifat ini adalah dominan, dengan kata lain gen tanaman yang tinggi mendominasi tanaman yang rendah. Untuk membantunya melakukan persilangan, Tania membuat sebuah program yang dapat membantunya menghitung rasio peluang munculnya tanaman dengan karakteristik tertentu apabila kedua "orang tua"-nya diketahui sifatnya.

Karena Tania baru bisa membuat program yang membaca bilangan, Tania harus membuat kode sendiri agar Ia dapat memasukkan karakteristik tanaman padi yang akan ia silangkan. Ia membuat aturan pemberian kode seperti berikut:

Pertama, Tania memberikan kode 0 sampai 2 untuk mewakili genotipe dari masing-masing fenotipe. Misalnya pada fenotipe Tinggi Tanaman, Tania memberikan nilai 2 untuk genotipe TT, nilai 1 untuk genotipe Tt, dan nilai 0 untuk genotipe tt.

Tabel 2.10 Kode Sifat Fenotipe dan Genotipe Tanaman Tania

Sifat	Fenotipe	Kode
Tinggi Tanaman	Tinggi (TT)	2
	Tinggi (Tt)	1
	Pendek (tt)	0
Jumlah Anakan	Banyak (AA)	2
	Banyak (Aa)	1
	Sedikit (aa)	0

Sifat	Fenotipe	Kode
Jumlah Bulir	Banyak (BB)	2
	Banyak (Bb)	1
	Sedikit (bb)	0
Ketahanan terhadap Penyakit	*Resisten* (RR)	2
	Resisten (Rr)	1
	Rentan (rr)	0

Kedua, Tania akan menuliskan masukan program berupa satu atau lebih nilai 0, 1, atau 2 yang merupakan sifat dari tanaman yang akan Ia silangkan. Angka di posisi pertama merupakan tinggi tanaman, posisi kedua merupakan jumlah anakan, posisi ketiga merupakan jumlah bulir, dan posisi keempat adalah ketahanan terhadap penyakit. Sebagai contoh, jika Tania akan memasukkan tanaman yang tinggi (TT), jumlah anakan sedikit (aa), jumlah bulir sedikit (bb), dan resisten terhadap penyakit (Rr), maka bilangan yang akan Tania masukkan adalah: 2 0 0 1.

Sekarang, kalian dapat mulai membantu Tania membuat program. Selamat mengerjakan!

Subproblem 1: Menghitung Persentase Suatu Sifat Keturunan Pertama (F1)

Deskripsi:

Pada subproblem ini, Tania hanya memperhatikan sifat Tinggi Tanaman dan Jumlah Anakan, dan ingin menghitung persentase dari suatu keturunan generasi pertama (F1).

Format Masukan:

Masukan berupa tiga baris. Dua baris pertama adalah sifat dari kedua orang tua, dan baris ketiga adalah sifat dari keturunan pertama yang akan dihitung persentase kemunculan fenotipenya.

Format Keluaran:

Keluaran berupa sebuah persentase fenotipe dari tanaman di baris ketiga yang ditulis sebagai bilangan pecahan dengan dua tempat desimal dan diakhiri dengan tanda %.

Tabel 2.11 Format Keluaran Program Persilangan Tanaman Subproblem 1

Contoh Masukan	Contoh Keluaran
2 2 0 0 0 0	25.00%

Penjelasan:

Pada contoh tersebut, Tania menyilangkan dua induk, yang bersifat tinggi (TT) dan banyak anakan (AA) dengan pendek (tt) dan sedikit anakan (aa). Lalu, Tania ingin menghitung persentase generasi pertama dengan sifat pendek (tt) dan sedikit anakan (aa). Rasio fenotipe dari persilangan ini adalah Tinggi dan Banyak Anakan : Pendek dan Sedikit Anakan = 3 : 1 = 75% : 25%.

Subproblem 2: Mengetahui Semua Sifat Keturunan Pertama (F1)

Deskripsi:

Pada subproblem ini, Tania hanya memperhatikan sifat Tinggi Tanaman dan Jumlah Anakan, dan ingin mengetahui sifat dari semua keturunan pertama (F1).

Format Masukan:

Masukan berupa dua baris yang berisi sifat dari kedua orang tua.

Format Keluaran:

Keluaran berupa 16 baris, masing masing baris berisi sifat dari semua keturunan pertama.

Tabel 2.12 Format Keluaran Program Persilangan Tanaman Subproblem 2

Contoh Masukan	Contoh Keluaran
2 2 1 1	Tinggi Banyak Tinggi Banyak Tinggi Banyak Tinggi Banyak Tinggi Banyak Tinggi Sedikit Tinggi Banyak Tinggi Sedikit Tinggi Banyak Tinggi Banyak Pendek Banyak Pendek Banyak Tinggi Banyak Tinggi Sedikit Pendek Banyak Pendek Sedikit

Penjelasan:

Pada contoh tersebut, Tania menyilangkan dua induk, yang bersifat tinggi (TT) dan banyak anakan (AA) dengan pendek (tt) dan sedikit anakan (aa). Lalu, Tania ingin menghitung persentase generasi kedua dengan sifat pendek (tt) dan sedikit anakan (aa). Rasio fenotipe dari persilangan ini adalah Tinggi dan Banyak Anakan : Pendek dan Sedikit Anakan = 15 : 1 = 93.75% : 6.25%.

Subproblem 3: Menghitung Persentase Suatu Sifat Keturunan Kedua (F2)

Deskripsi:

Pada *subproblem* ini, Tania akan memperhatikan keempat sifat tanaman padi tersebut, dan ingin menghitung persentase dari suatu keturunan generasi kedua (F2).

Format Masukan:

Masukan berupa tiga baris. Dua baris pertama adalah sifat dari kedua orang tua, dan baris ketiga adalah sifat dari keturunan kedua yang akan dihitung persentase kemunculan fenotipenya.

Format Keluaran:

Keluaran berupa sebuah persentase fenotipe dari tanaman di baris ketiga yang ditulis sebagai bilangan pecahan dengan dua tempat desimal dan diakhiri dengan tanda %.

Tabel 2.13 Format Keluaran Program Persilangan Tanaman Subproblem 3

Contoh Masukan	Contoh Keluaran
2 2 2 2 0 0 0 0 0 0 0 0	0.39%

Penjelasan:

Pada contoh tersebut, Tania menyilangkan dua induk, yang bersifat:

- 1: Tinggi (TT), Banyak Anakan (AA), Banyak Bulir (BB), dan Resisten (RR)
- 2: Pendek (tt), Sedikit Anakan (aa), Sedikit Bulir (bb), dan Rentan (rr)

Lalu, Tania ingin menghitung persentase generasi kedua dengan sifat:

Pendek (tt), Sedikit Anakan (aa), Sedikit Bulir (bb), dan Rentan (rr). Persentase fenotipe generasi kedua dari sifat tersebut adalah sama dengan 0.390625% yang jika dibulatkan ke atas menjadi 0.39%.

Ayo Merancang Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-14-U: Merancang Algoritma Persilangan Tanaman

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang persilangan tanaman yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam Buku Kerja kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Ayo Buat Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-15: Membuat Program Persilangan Tanaman

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan

menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Ini Berpikir Komputasional!

Pada Kelas X, kalian telah belajar bahwa Informatika dapat diterapkan di berbagai bidang keilmuan. Aktivitas di atas adalah salah satu contoh penerapan Informatika di bidang Biologi. Pada saat mengerjakan aktivitas di atas, kalian memanfaatkan pengetahuan kalian di pemrograman dan juga biologi untuk menyelesaikan permasalahan yang diberikan. Pada aktivitas ini juga kalian merasakan pengalaman bekerja berpasangan dalam membuat program. Membuat program yang kompleks seringkali melibatkan suatu tim yang harus bekerja sama dan saling melengkapi satu sama lain. Yang kalian lakukan di aktivitas ini adalah bagaimana kalian sebagai tim dapat saling mengecek hasil pekerjaan untuk menghasilkan program yang benar dan berkualitas tinggi. Pada aktivitas ini juga, kalian telah merasakan bagaimana Berpikir Komputasional diterapkan dalam pemrograman. Kalian dapat mencoba mengidentifikasi prinsip Berpikir Komputasional apa saja yang kalian terapkan dalam menyelesaikan problem tersebut.

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja , dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

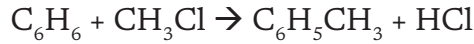
1. Apa yang kalian rasakan saat membuat suatu algoritma dan program dari suatu permasalahan di mata pelajaran lain yang telah kalian ikuti?
2. Apakah program yang kalian buat dapat membantu kalian atau orang lain?
3. Apakah solusi yang telah kalian buat dapat dimanfaatkan untuk menyelesaikan masalah lain yang sejenis?
4. Adakah pengembangan lebih lanjut (enhancing) yang terpikir oleh kalian agar program menjadi lebih bermanfaat?
5. Apa yang kalian rasakan saat memeriksa solusi algoritma dan program teman kalian?
6. Apa yang kalian rasakan saat solusi algoritma dan program kalian diperiksa oleh teman kalian?
7. Prinsip Berpikir Komputasional apa saja yang kalian terapkan saat menyelesaikan aktivitas di atas?
8. Pelajaran paling berkesan apa yang kalian dapatkan dari latihan ini?

4. *Problem* Stoikiometri

Stoikiometri adalah ilmu yang mempelajari dan menghitung hubungan kuantitatif dari reaktan dan produk dalam reaksi kimia. Pada bagian ini kalian akan diajak untuk membangun sebuah program yang dapat membantu kalian untuk dapat memproses masukan berupa suatu persamaan reaksi kimia. Manusia melakukan observasi pada reaksi kimia pada alam, dan dengan menggunakan berbagai hukum seperti hukum kekekalan massa, hukum perbandingan tetap, dan hukum perbandingan berganda, melahirkan suatu model reaksi kimia. Salah satunya adalah stoikiometri.

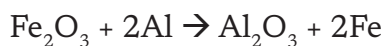
Setelah memahami stoikiometri, kalian dapat membaca sebuah persamaan reaksi kimia dan mengetahui bahwa persamaan tersebut benar karena semua pereaksi dikonsumsi

(habis bereaksi), tidak ada kekurangan pereaksi, dan tidak ada kelebihan pereaksi (sisa). Sebelum masuk ke *problem* yang akan dikerjakan, kalian dapat mencoba menuliskan secara struktur proses yang kalian lakukan untuk mengecek persamaan reaksi berikut apakah benar atau tidak.

$$C_6H_6 + CH_3Cl \rightarrow C_6H_5CH_3 + HCl$$

Problem: "Simulasi Stoikiometri"

Tujuan kalian dalam *problem* ini adalah membuat program yang dapat mengecek benar tidaknya persamaan reaksi kimia yang diberikan. Program yang dapat mengecek seluruh kemungkinan persamaan kimia sangat kompleks. Oleh karena itu, kita memerlukan batasan-batasan yang dapat menyederhanakan masalah tersebut, yaitu:

- Penulisan unsur kimia pada masukan sudah sesuai dengan aturan. Misal: Li untuk litium dan H untuk hidrogen.
- Persamaan reaksi yang diproses tidak melibatkan senyawa ionik dan senyawa yang memiliki tanda kurung.
- Persamaan reaksi yang diproses dijamin benar.

Selain itu, aturan penulisan yang menggunakan IUPAC seperti pada persamaan reaksi kimia berikut akan disederhanakan. Sebagai contoh, suatu persamaan reaksi kimia berikut

$$Fe_2O_3 + 2Al \rightarrow Al_2O_3 + 2Fe$$

akan kita tulis dalam *problem* ini sebagai berikut:

```
1 Fe 2 O 3
2 Al 1
>
1 Al 2 O 3
1 Fe 2
```

Jika kalian tertarik, kalian bisa saja membuat program yang dapat membaca masukan berupa persamaan reaksi kimia seperti yang biasa dituliskan dengan mempelajari konsep parsing. Parsing adalah sebuah proses menganalisis *string* yang mengikuti aturan formal tertentu sehingga *string* tersebut dapat dibaca oleh program. Dalam hal ini, aturan formal yang diikuti adalah aturan formal penulisan reaksi kimia. Parsing tidak dibahas pada buku ini, namun kalian dapat mencari tahu lebih lanjut tentang konsep parsing dari berbagai sumber lainnya.

Subproblem 1: Menghitung Atom

Deskripsi:

Pada *subproblem* ini, kalian harus membuat program untuk menghitung banyaknya atom unsur tertentu yang terdapat pada senyawa kimia yang diberikan. Program kalian akan membaca sebuah senyawa dengan format yang telah ditentukan di bagian sebelumnya, kemudian mencetak banyaknya atom dari masing-masing unsur penyusun senyawa tersebut.

Format Masukan:

Sebuah senyawa yang telah diformat sesuai dengan deskripsi *problem*.

Format Keluaran:

Untuk setiap unsur, cetak lambang dari unsur tersebut dan banyaknya atom yang menyusun unsur tersebut. Kedua nilai ini dipisahkan oleh satu karakter spasi dan diakhiri dengan *newline*. Lambang unsur dan banyaknya atom dicetak berurutan mulai dari unsur paling kiri pada penulisan senyawa.

Tabel 2.14 Format Keluaran Program Stoikiometri *Subproblem* 1

Contoh Masukan	Contoh Keluaran
1 C 1 H 3 Cl 1	C 1 H 3 Cl 1
2 H 2 O 1	H 4 O 2

Penjelasan:

Senyawa pada contoh pertama adalah CH_3CL dan contoh kedua adalah $2H_2O$.

Subproblem 2: Mengecek Persamaan Reaksi

Deskripsi:

Pada *subproblem* ini, kalian akan membuat program yang akan membaca sebuah persamaan reaksi kimia, dan menentukan apakah persamaan reaksi kimia tersebut setimbang. Suatu atom dijamin hanya akan muncul 1 kali pada reaktan dan 1 kali pada produk. Apabila setimbang, maka program cukup mengeluarkan pesan "Reaksi Setimbang". Apabila persamaan tersebut tidak setimbang, maka program akan mengeluarkan pesan "Reaksi Tidak Setimbang" dan menyajikan banyak atom dari reaktan yang kurang atau berlebih.

Format Masukan:

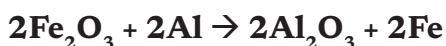
Sebuah persamaan reaksi kimia dengan format yang telah dijelaskan pada bagian penjelasan *problem*. Untuk mengakhiri masukan, kalian akan menginput sebuah karakter "F".

Format Keluaran:

Jika persamaan setimbang, cetak "Reaksi Setimbang". Jika reaksi tidak setimbang, program akan mencetak lambang dari unsur yang kurang atau berlebih beserta

dengan selisih jumlah atomnya. Unsur dicetak berurutan mulai dari unsur yang paling kiri. Untuk setiap unsur tersebut, cetak dalam satu baris dengan format Lambang Unsur: jumlah atom. Jumlah atom diberi tanda negatif jika kurang, dan diberi tanda positif jika berlebih.

Contoh:

Misalnya program menerima masukan berupa persamaan reaksi kimia berikut:

$$2Fe_2O_3 + 2Al \rightarrow 2Al_2O_3 + 2Fe$$

Program akan melihat bahwa persamaan tersebut tidak setimbang dan akan mengeluarkan keluaran berikut:

Tabel 2.15 Format Keluaran Program Stoikiometri *Subproblem* 2

Contoh Masukan	Contoh Keluaran
2 Fe 2 O 3 2 Al 1 > 2 Al 2 O 3 2 Fe 1 F	Reaksi Tidak Setimbang Fe:+2 Al:-2

Subproblem 3: Menghitung Hasil Reaksi

Deskripsi:

Pada *subproblem* ini, program akan menerima masukan berupa persamaan reaksi kimia. Bagian reaktan dari persamaan tersebut sudah ditulis dengan koefisien yang lengkap. Namun, koefisien dari bagian produk persamaan tersebut belum lengkap. Suatu atom dijamin hanya akan muncul 1 kali pada reaktan dan 1 kali pada produk. Setelah membaca persamaan tersebut, program akan mencetak koefisien dari setiap molekul di bagian produk.

Format Masukan:

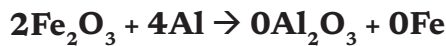
Sebuah persamaan reaksi kimia dengan format yang telah dijelaskan pada bagian penjelasan *problem*, akan

tetapi koefisien pada seluruh molekul produk diberi nilai 0. Untuk mengakhiri masukan, kalian akan menginput sebuah karakter “F”.

Format Keluaran:

Keluaran dari program adalah koefisien dari setiap molekul produk yang dihasilkan dari persamaan tersebut. Koefisien tersebut harus membuat persamaan reaksi yang setimbang. Jika bagian produk terdiri atas lebih dari satu molekul, koefisien dicetak dari molekul produk yang paling kiri dan dipisahkan dengan karakter spasi.

Contoh:

Misalnya program menerima masukan berupa persamaan reaksi kimia berikut:

$$\mathbf{2Fe_2O_3 + 4Al \rightarrow 0Al_2O_3 + 0Fe}$$

Program akan melihat bahwa persamaan tersebut tidak setimbang dan akan mengeluarkan keluaran berikut:

Tabel 2.16 Format Keluaran Program Stoikiometri *Subproblem* 3

Contoh Masukan	Contoh Keluaran
2 Fe 2 O 3 4 Al 1 > 0 Al 2 O 3 0 Fe 1 F	2 4

Dengan koefisien 2 dan 4 pada sisi produk, maka persamaan reaksi kimia akan setimbang menjadi

2Fe2O3 + 4Al → 2Al2O3 + 4Fe.

Subproblem 4: Melengkapi Koefisien Reaktan dan Produk

Deskripsi:

Pada *subproblem* ini, persamaan reaksi kimia yang dibaca seluruhnya memiliki koefisien 0. Program harus membaca persamaan reaksi kimia tersebut dan mencetak koefisien setiap molekul sedemikian sehingga reaksi setimbang. Karena kemungkinan koefisien ini sangat banyak, kalian hanya perlu mencetak kemungkinan koefisien yang paling kecil. Terdapat tepat 2 senyawa pada reaktan, dan 2 senyawa pada hasil dan masing-masing atom hanya muncul tepat 1 kali pada reaktan dan 1 kali pada hasil.

Format Masukan:

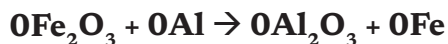
Sebuah persamaan reaksi kimia dengan format yang telah dijelaskan pada bagian penjelasan problem, akan tetapi koefisien pada seluruh molekul reaktan dan produk diberi nilai 0.

Format Keluaran:

Keluaran program adalah persamaan reaksi kimia yang telah dilengkapi dengan koefisien terkecil yang membuat persamaan reaksi tersebut setimbang.

Contoh:

Misalnya program menerima masukan berupa persamaan reaksi kimia berikut:

$$0Fe_2O_3 + 0Al \rightarrow 0Al_2O_3 + 0Fe$$

Program akan mencetak persamaan reaksi.

Tabel 2.17 Format Keluaran Program Stoikiometri Subproblem 4

Contoh Masukan
1 Fe 2 O 3
2 Al 1
>
1 Al 2 O 3
2 Fe 1

Ayo Merancang Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-16-U: Merancang Algoritma Simulasi Stoikiometri

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang stoikiometri yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam **Buku Kerja** kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Ayo Buat Program!

Aktivitas Berpasangan

Aktivitas SAP-K11-17: Membuat Program Simulasi Stoikiometri

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Ini Berpikir Komputasional!

Pada problem stoikiometri ini, selain kalian membuat program yang mensimulasikan suatu aturan di bidang Kimia, kalian pun belajar membuat program yang menerima masukan yang dekat dengan bahasa yang digunakan manusia.

Ayo, Kita Renungkan

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apa yang kalian rasakan saat membuat suatu algoritma dan program dari suatu permasalahan di mata pelajaran lain yang telah kalian ikuti?

2. Apakah program yang kalian buat dapat membantu kalian atau orang lain?
3. Apakah solusi yang telah kalian buat dapat dimanfaatkan untuk menyelesaikan masalah lain yang sejenis?
4. Adakah pengembangan lebih lanjut (*enhancing*) yang terpikir oleh kalian agar program menjadi lebih bermanfaat?
5. Apa yang kalian rasakan saat memeriksa solusi algoritma dan program teman kalian?
6. Apa yang kalian rasakan saat solusi algoritma dan program kalian diperiksa oleh teman kalian?
7. Prinsip Berpikir Komputasional apa saja yang kalian terapkan saat menyelesaikan aktivitas di atas?
8. Pelajaran paling berkesan apa yang Kalian dapatkan dari latihan ini?

E. Proyek Praktik Lintas Bidang (PLB)

Pertama-tama, sesuai dengan petunjuk dari guru, buatlah kelompok yang masing-masing terdiri dari 2 orang peserta didik (jika kelas berjumlah ganjil, satu kelompok boleh berisi 3 orang peserta didik). Saat mengerjakan proyek ini, diperlukan kerja sama antar anggota kelompok dengan baik untuk menghasilkan jawaban dan solusi yang tepat.

Tujuan utama dari proyek lintas bidang ini adalah menghasilkan sebuah program (dalam Bahasa C/C++) yang dapat menyelesaikan permasalahan *knapsack* (baik yang berjenis rational *knapsack* maupun 0-1 *knapsack*) dengan menerapkan prinsip-prinsip penyelesaian masalah yang telah dipelajari pada bagian Berpikir Komputasional. Hasil akhir yang diharapkan adalah berupa dua hal:

- Laporan analisis dan perancangan program
- Kode program yang dapat dijalankan dan memberikan solusi dari setiap masukan yang diberikan

Dalam pengerjaan ini, kalian akan dipandu dengan serangkaian langkah-langkah. Ikutilah langkah-langkah tersebut dan kerjakan bersama rekan satu tim untuk mendapatkan hasil terbaik.

1. Penjelasan Permasalahan

Permasalahan yang akan diselesaikan pada PLB ini adalah salah contoh dari sebuah permasalahan optimasi. Dalam dunia informatika, optimasi adalah suatu tipe permasalahan dimana kita ingin mencari nilai yang optimal (paling baik) untuk sebuah kasus permasalahan tertentu. Konsep "paling baik" dapat berarti dua hal: paling kecil (minimal) atau paling besar (maksimal), sehingga secara umum ada dua jenis permasalahan optimasi, yaitu optimasi untuk mencari nilai maksimum dan optimasi untuk mencari nilai minimum. Nilai atau besaran yang ingin dioptimalkan (minimal atau maksimal) biasa disebut sebagai fungsi tujuan (fungsi obyektif).

Pada umumnya, ketika mencari suatu nilai yang optimal dalam sebuah permasalahan optimasi, kita harus menghadapi suatu kendala yang mengatur pilihan-pilihan yang dapat kita ambil. Mungkin, tidak semua pilihan dapat kita ambil. Ketika kita melakukan optimasi, kita harus memilih di antara pilihan-pilihan tersebut mana yang memberikan nilai fungsi obyektif yang paling optimal.

Sebagai contoh, kita tinjau kembali permasalahan penukaran uang yang telah kita pelajari pada bagian Berpikir Komputasional. Pada permasalahan tersebut, kita diberikan beberapa pecahan uang (misalnya: 100 Rupiah, 200 Rupiah, 500 Rupiah, 1000 Rupiah, 5000 Rupiah dan 10.000 Rupiah), serta sebuah besaran nilai uang yang ingin kita tukarkan, (misalnya Rp 27.800). Kita diminta untuk menentukan, berapa banyaknya pecahan uang terkecil yang dapat kita tukarkan untuk mencapai besaran tersebut?

Permasalahan ini adalah sebuah permasalahan optimasi dimana jenis optimasi pada permasalahan ini adalah pencarian

nilai minimum (terkecil). Fungsi tujuannya adalah banyaknya pecahan uang yang harus dikumpulkan, sedangkan kendala yang diberikan adalah besaran-besaran pecahan uang yang tersedia, serta jumlah total besaran nilai uang yang harus dikumpulkan.

Pada PLB ini kita akan mempelajari bagaimana menyelesaikan permasalahan yang biasa disebut sebagai knapsack problem. Berikut ini adalah deskripsi umum permasalahan knapsack problem.

Terdapat beberapa buah barang yang dapat kita ambil. Setiap barang memiliki dua hal: **bobot** dan **nilai**. Kita diberikan sebuah wadah yang dapat menampung beberapa buah barang, namun kapasitasnya berat beban yang dapat ditampungnya terbatas sampai suatu nilai tertentu (tidak boleh mengisi wadah melebihi batas maksimal kapasitas beban ini, namun tentunya boleh kurang). Pertanyaannya adalah: "Dapatkah kita memilih barang-barang yang diambil untuk dimasukkan ke dalam wadah, sedemikian rupa sehingga kita mendapatkan total nilai yang sebesar-besarnya, namun tidak melewati batas maksimal beban wadah tersebut?"

Sebagai contoh, perhatikanlah kasus berikut ini. Misalnya kita memiliki 5 buah barang, sebut saja namanya A, B, C, D dan E, dengan bobot (dalam kg) dan nilai masing-masing sebagai berikut:

Tabel 2.18 Keterangan Bobot dan Nilai dari 5 Barang (*Knapsack Problem*) Kasus 1

Barang	A	B	C	D	E
Bobot	5	4	7	8	10
Nilai	10	5	7	12	8

Misalkan kapasitas maksimal dari wadah adalah 20 kg. Maka kita dapat melakukan beberapa contoh pilihan sebagai berikut:

- Ambil barang A, B dan D. Total bobot yang harus ditampung pada wadah adalah 5 + 4 + 8 = 17 kg, yang

masih di bawah maksimal kapasitas wadah. Total nilai yang kita dapatkan adalah 10 + 5 + 12 = 27.

- Ambil barang B, C dan D, dengan total bobot = 4 + 7 + 8 = 19 kg, dengan total nilai yang didapatkan = 5 + 7 + 12 = 24.
- Ambil barang A, C dan D, dengan total bobot = 5 + 7 + 8 = 20 kg, dengan total nilai yang didapatkan = 10 + 7 + 12 = 29.

Tentunya kita tidak dapat mengambil A, B, C dan D sekaligus misalnya, karena total bobotnya adalah 5 + 4 + 7 + 8 = 24 > 20 kg, sehingga akan melewati kapasitas dari wadah. Ternyata, nilai total 29 yang didapatkan dari memilih A, C dan D di atas adalah nilai maksimal yang kita dapatkan, dan tidak ada pemilihan pengambilan barang yang lain yang menghasilkan nilai total > 29 dengan tetap memenuhi persyaratan total bobot < kapasitas.

2. *Rational* dan 0-1 *Knapsack*

Terdapat dua jenis variasi dari permasalahan *knapsack*. Yang pertama adalah yang disebut sebagai *rational knapsack*. Pada permasalahan ini, setiap barang dapat dianggap sebagai barang dapat dipecah, artinya, boleh diambil sebagian saja (tidak harus semuanya). Contoh barang seperti ini adalah misalnya: air, minyak, beras, pasir, dan lain-lain. Jika Anda memiliki 10 kg minyak, Anda dapat memilih untuk mengambil hanya 7 kg dari minyak tersebut, sehingga nilai yang Anda dapatkan adalah 7/10 dari nilai total keseluruhan minyak. Karena setiap barang sifatnya dapat dipecah dan diambil sebagian, maka permasalahan rational *knapsack* biasanya lebih mudah diselesaikan, karena kita dapat mengatur untuk setiap barang, berapa bagian yang ingin kita ambil.

Variasi kedua dari permasalahan *knapsack* adalah yang disebut sebagai 0–1 *knapsack*. Sesuai dengan namanya, pada permasalahan 0–1 *knapsack*, kita hanya dapat memilih untuk mengambil atau tidak mengambil setiap barang, karena setiap barang sifatnya adalah tunggal dan tidak dapat

dipecah belah. Contoh barang seperti ini adalah misalnya: laptop, sepeda, mobil, hewan dan sebagainya, yang tentunya tidak dapat diambil hanya sebagian/sepotong saja, dan kita harus menentukan apakah setiap barang akan diambil atau tidak. Permasalahan 0-1 *knapsack* biasanya lebih sulit untuk diselesaikan daripada *rational knapsack*.

Perhatikan contoh kasus berikut ini. Diberikan tiga buah barang yang tidak dapat dibagi/dipecah, A, B dan C dengan bobot dan nilai sebagai berikut yang akan ditaruh dalam sebuah wadah.

Tabel 2.19 Keterangan Bobot dan Nilai dari 3 Barang (Knapsack Problem) Kasus 2

Barang	A	B	C
Bobot	5	4	7
Nilai	10	5	7

Misalkan bahwa kapasitas wadah adalah 10 kg. Apabila kita mengasumsikan bahwa jenis permasalahan ini adalah *rational knapsack*, maka kita dapat mengambil solusi sebagai berikut:

- Ambil semua barang A, sehingga didapatkan bobot = 5 kg, total nilai sementara = 10.
- Kemudian ambil semua barang B, sehingga didapatkan bobot = 5 + 4 = 9kg, dan nilai total sementara = 10 + 5 = 15.
- Ambil 1/7 bagian dari barang C, sehingga didapatkan total bobot = 9 + 1/7 * 7 = 10 kg, dan nilai total menjadi 15 + 1/7 * 7 = 16.

Hasil ini ternyata adalah hasil yang maksimal, kita tidak dapat memperoleh nilai total yang lebih besar dari 16 tanpa melebihi kapasitas wadah, yaitu 10 kg.

Nah sekarang, seandainya kita menganggap permasalahan ini sebagai permasalahan 0-1 knapsack, maka kita hanya dapat mengambil setiap barang secara keseluruhan, atau tidak sama sekali. Maka dalam kasus ini, solusi optimalnya adalah

dengan mengambil barang A dan B saja, dengan total bobot = 5 + 4 = 9 < 10 kg, dan total nilai = 10 + 5 = 15.

Aktivitas PLB

Aktivitas Berpasangan

Aktivitas SAP-K11-18-U: Memahami Permasalahan *Knapsack*

Jawablah pertanyaan-pertanyaan berikut dan tuliskan dalam laporan analisis.

1. Apakah jenis optimasi pada permasalahan *knapsack*? Apakah mencari minimum, ataukah maksimum? Jelaskan!
2. Tentukan apa yang menjadi fungsi tujuan dari permasalahan *knapsack*!
3. Tentukan apa yang menjadi kendala pada optimasi untuk permasalahan *knapsack*!
4. Perhatikan permasalahan knapsack yang ditunjukkan oleh Tabel 2.20 berikut. Diberikan 6 buah barang, A, B, s/d F dengan bobot dan nilai sebagai berikut:

Tabel 2.20 Keterangan Bobot dan Nilai dari 6 Barang (Knapsack Problem) Aktivitas PLB

Barang	A	B	C	D	E	F
Bobot	3	8	5	4	10	8
Nilai	6	4	5	6	5	10

Asumsikan bahwa tas memiliki kapasitas maksimal = 24 kg.

a. Apakah pilihan mengambil barang-barang B, D, E dan F diperbolehkan sebagai solusi sesuai dengan kendala optimasi pada permasalahan tersebut? Mengapa?

b. Apakah pilihan mengambil barang-barang A, D, E diperbolehkan sebagai solusi sesuai dengan kendala optimasi pada permasalahan tersebut? Apakah fungsi tujuan mencapai nilai optimal dengan memilih A, D dan E saja? Mengapa?

5. Tentukan jawaban permasalahan knapsack tersebut pada soal no. 4, jika menggunakan variasi permasalahan rational *knapsack*!
6. Pada soal no. 4, apakah solusinya, jika digunakan variasi 0-1 knapsack? Apakah sama dengan solusi untuk variasi rational *knapsack*?

Aktivitas PLB

Aktivitas Berpasangan

Aktivitas SAP-K11-19: Mengkodekan Permasalahan *Knapsack*

Membuat Skema Masukan Program

Sebelum kita dapat menyelesaikan permasalahan di atas dengan menggunakan sebuah program, hal pertama yang perlu kita lakukan adalah menentukan bagaimana menyatakan sebuah kasus permasalahan *knapsack* menjadi sebuah bentuk/format yang mudah dibaca dan diolah oleh komputer. Cara yang paling mudah adalah dengan menuliskan besaran-besaran yang ada pada permasalahan tersebut dan menyatakannya dalam *array-array* atau deretan bilangan yang sesuai. Kita dapat memulai dengan menuliskan banyaknya barang yang tersedia, kemudian diikuti dengan bobot masing-masing barang, kemudian diikuti dengan nilai masing-masing barang, dan terakhir kita tuliskan kapasitas maksimal dari tas. Sebagai contoh, untuk kasus *knapsack* dimana barang-barang yang akan dimasukkan pada tas ditunjukkan pada Tabel 2.21 berikut:

Tabel 2.21 Keterangan Bobot dan Nilai dari 5 Barang (Knapsack Problem) Aktivitas PLB

Barang	A	B	C	D	E
Bobot	5	4	7	8	10
Nilai	10	5	7	12	8

Dan dimana jumlah maksimal bobot yang dapat ditampung adalah 20 kg, maka kita dapat merepresentasikan permasalahan ini menjadi bentuk sebagai berikut:

Contoh masukan 1

5

5 4 7 8 10

10 5 7 12 8

20

Sekarang perhatikan permasalahan pada Tabel 2.22 berikut:

Tabel 2.22 Keterangan Bobot dan Nilai dari 6 Barang (Knapsack Problem) Aktivitas PLB

Barang	A	B	C	D	E	F
Bobot	3	8	5	4	10	8
Nilai	6	4	5	6	5	10

Dimana kapasitas maksimal yang dapat ditampung adalah 25 kg. Sekarang perhatikan juga skema masukan pada contoh berikut:

Contoh masukan 2

8

3 10 6 7 9 10 7 5

1 10 8 1 7 8 9 18

35

Jawablah pertanyaan-pertanyaan berikut pada lembar jawaban/laporan PLB!

Pertanyaan:

1. Tentukan pengkodean yang sesuai untuk permasalahan yang ditunjukkan pada Tabel 2.22 di atas, dengan menggunakan skema pengkodean yang telah dijelaskan. Jelaskan!
2. Tuliskan/jelaskan pada laporan analisis kalian, deskripsi permasalahan untuk contoh masukan 2 di atas!
3. Jelaskan dalam satu paragraf, skema pengkodean di atas, agar dapat dipahami oleh orang lain!

4. Mungkinkah sebuah representasi data tidak valid/tidak sesuai? Berikan contohnya, dan tuliskan penjelasannya pada laporan analisis!

Membaca Masukan dan Menyimpan Data

Sekarang, setelah kita memiliki sebuah representasi data yang sesuai untuk setiap permasalahan, marilah kita memulai untuk membuat program yang dapat menyelesaikan permasalahan yang telah dinyatakan dengan skema pengkodean tersebut. Pada bagian ini, kita akan membuat program yang membaca masukan dengan format di atas, kemudian hanya menampilkan kembali deskripsi soal sebagai keluaran. Perhatikan contoh di bawah ini.

Misalkan diberikan masukan berupa data sebagaimana ditunjukkan pada Contoh Masukan 1. Maka program harus mengeluarkan output sebagai berikut:

```
Banyak barang = 5
Barang 1: bobot = 5 kg, nilai = 10
Barang 2: bobot = 4 kg, nilai =  5
...
Barang 5: bobot = 10 kg, nilai = 8
Kapasitas maksimal = 20 kg
```

Untuk menyelesaikan permasalahan ini, kita perlu mengingat kembali konsep *array* pada Bagian Algoritma dan Pemrograman, dan menggunakannya untuk menyimpan data bobot dan nilai dari barang-barang yang tersedia.

Aktivitas PLB

Aktivitas Berpasangan

Aktivitas SAP-K11-20-U: Merancang Algoritma Penyelesaian Masalah Knapsack

Pada bagian Berpikir Komputasional, kita sudah mempelajari beberapa jenis teknik penyelesaian masalah. Untuk permasalahan

knapsack pada PLB ini, kita harus menentukan strategi penyelesaian masalah yang manakah yang sesuai untuk diterapkan.

1. Untuk permasalahan *rational knapsack*, tentukan apakah strategi *greedy* ataukah *dynamic programming* yang sesuai untuk diterapkan? Jelaskan pada laporan analisis kamu, bagaimana strategi *greedy* atau *dynamic programming* dapat diterapkan pada permasalahan rational *knapsack*!
2. Untuk permasalahan 0–1 *knapsack*, tentukan apakah strategi *greedy* ataukah *dynamic programming* yang sesuai untuk diterapkan?
3. Tuliskan dalam notasi *pseudocode* algoritma yang sesuai untuk menyelesaikan permasalahan *rational knapsack* menggunakan strategi yang Anda pilih pada bagian nomor 1!
4. **[Opsional]** Tuliskan dalam notasi *pseudocode* algoritma yang sesuai untuk menyelesaikan permasalahan 0-1 *knapsack* menggunakan strategi yang Anda pilih pada bagian nomor 2!

Aktivitas PLB

Aktivitas Berpasangan

Aktivitas SAP-K11-21: Mengimplementasikan dan Menguji Program Solusi Knapsack

Implementasi program

Pada bagian ini, setiap kelompok akan membuat program yang menyelesaikan permasalahan *rational knapsack*. Program yang dibuat memiliki spesifikasi (ketentuan) sebagai berikut:

1. Program membaca masukan berupa representasi pengkodean kasus sebagaimana dijelaskan di atas.
2. Kemudian program tersebut menjalankan strategi penyelesaian program sebagaimana telah dirancang pada *pseudocode* pada bagian sebelumnya.
3. Program kemudian menghasilkan keluaran berupa sebuah angka yang menunjukkan berapa nilai total terbesar yang

dapat dicapai (nilai optimal) untuk kasus yang diberikan pada masukan.

Misalnya, jika program diberikan masukan sebagaimana contoh masukan 1, maka program akan memberikan keluaran sebagaimana dijelaskan pada Contoh 1, yaitu 29.

Contoh Masukan	Contoh Keluaran
5 5 4 7 8 10 10 5 7 12 8 20	2 9

Untuk melakukan implementasi program ini, dianjurkan setiap kelompok melakukan sistem yang disebut sebagai *pair programming*. *Pair programming* adalah sebuah teknik pemrograman dimana dua orang bekerja bersama dalam menghasilkan sebuah program. Satu orang bertindak untuk mengerjakan penulisan program, satu orang lagi bertindak untuk mengawasi dan memberikan masukan terhadap penulisan program rekannya, sekaligus memikirkan strategi untuk penulisan program berikutnya. Selanjutnya kedua orang dapat bergantian peran, apabila diperlukan.

Jika kalian mengimplementasikan teknik *greedy* untuk menyelesaikan permasalahan *rational knapsack*, maka kalian akan memerlukan proses pengurutan data [Ingat kembali materi Berpikir Komputasional di Kelas X!]. Untuk melakukan hal ini, kalian dapat langsung saja menggunakan fitur dalam bahasa C++ yang menyediakan pustaka untuk melakukan pengurutan, yaitu dengan fungsi `sort()`. Jika kalian memiliki sebuah `std::vector<>` bernama data misalnya, maka kalian tinggal memanggil fungsi `sort()` sebagai berikut: `sort(data.begin(), data.end())`. Maka data pada *vector* tersebut akan terurut. Lebih lanjut, kita dapat menggunakan proses pengurutan tidak hanya pada satu data, tapi kita dapat juga menggunakan hasil pengurutan untuk mengurutkan

data yang lain. Perhatikan contoh berikut, dimana kita ingin menampilkan data nama-nama peserta didik sesuai dengan urutan nilai ujiannya, secara terurut menaik!

```
1:  /* Melakukan pengurutan pada dua buah vector */
2:  #include <stdio.h>
3:  #include <vector>
4:  #include <string>
5:  #include <algorithm>
6:
7:  int main()
8:  {
9:      std::vector<int> nilai {90, 60, 80, 75};
10:     std::vector<std::string>
11:         nama {"Andi", "Budi", "Cici", "Didi"};
12:
13:     std::vector<int> index;
14:     for(int i=0; i<4; i++) index.push_back(i);
15:
16:     // Urutkan index berdasarkan nilai
17:     std::sort(index.begin(), index.end(),
18:        [&nilai](int index1, int index2)
19:        { return nilai[index1] < nilai[index2]; });
20:
21:     // Cetak nama berdasarkan urutan nilai
22:     for(int i=0; i<4; i++)
23:         printf("%s ", nama[index[i]].c_str());
24:
25:     return 0;
26: }
```

Keluaran dari program tersebut adalah:

Budi Didi Cici Andi

Jika kita ingin mengurutkan secara menurun, kalian dapat merubah operasi perbandingan < pada baris berikut:

```
return nilai[index1] < nilai[index2];
```

Menjadi operasi perbandingan: >

```
return nilai[index1] > nilai[index2];
```

Untuk lebih jelasnya, dengarkan penjelasan gurumu mengenai hal ini!

Pengujian program

Setelah program diimplementasikan, langkah terakhir adalah melakukan pengujian. Pada tahap ini, setiap kelompok

membuat beberapa kasus uji. Setiap kasus uji berisi kedua komponen data, yaitu masukan dan keluaran yang telah diperiksa secara manual oleh kedua anggota kelompok, untuk memastikan kebenarannya. Setiap kelompok perlu membuat minimal 5 kasus uji yang panjangnya masing-masing berisi minimal 5 buah barang.

Selanjutnya, program diuji dengan kasus-kasus uji tersebut, dan diperiksa apakah keluaran dari program sesuai dengan keluaran yang diinginkan. Untuk menambah keketatan pengujian, setiap kelompok dapat "meminjam" kasus-kasus uji dari 2 kelompok lain untuk melakukan pengujian program masing-masing.

Tuliskan hasil pengujian keseluruhan pada laporan akhir PLB! Ikuti format berikut untuk melaporkan hasil pengujian.

Tabel 2.23 Format Hasil Pengujian Program Laporan Akhir PLB

Masukan	Keluaran yang diharapkan	Keluaran yang didapatkan	Hasil Pengujian
5 5 4 7 8 10 10 5 7 12 8 20	29	29	Sesuai
3 5 4 7 10 5 7	16	15	Tidak sesuai
....	....	....	....

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apa yang kalian rasakan saat membuat suatu program di Praktik Lintas Bidang ini?
2. Apakah program yang kalian buat dapat membantu kalian atau orang lain?
3. Apakah solusi yang telah kalian buat dapat dimanfaatkan untuk menyelesaikan masalah lain yang sejenis?
4. Adakah pengembangan lebih lanjut (*enhancing*) yang terpikir oleh kalian agar program menjadi lebih bermanfaat?
5. Apa yang kalian rasakan saat memeriksa solusi algoritma dan program teman kalian?
6. Apa yang kalian rasakan saat solusi algoritma dan program kalian diperiksa oleh teman kalian?
7. Pelajaran paling berkesan apa yang kalian dapatkan dari aktivitas latihan ini?

KEMENTERIAN PENDIDIKAN,
KEBUDAYAAN, RISET, DAN
TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas XI
Penulis : Paulina H. Prima Rosa
ISBN: 978-602-244-861-7

Berpikir Kritis dan Dampak Sosial Informatika

Tujuan Pembelajaran

Setelah mempelajari bab ini kalian akan mengenal dan belajar mengkritisi berbagai contoh penerapan Informatika khususnya dalam bidang pertanian dan kesehatan. Nah, untuk itu, pertama-tama kalian juga perlu memahami pentingnya berpikir kritis, membangun kebiasaan berpikir kritis, dan mempergunakan kemampuan berpikir kritis untuk mengkaji penerapan informatika dalam bidang pertanian dan kesehatan. Hasil analisis ini akan membantu kalian menentukan sikap yang tepat dalam menghadapi persoalan-persoalan yang mungkin muncul berkaitan dengan penerapan informatika dalam berbagai bidang tersebut kali ini. Selanjutnya, kalian

dapat memakai pola berpikirnya untuk persoalan lainnya. Selain itu, kemampuan berpikir kritis akan bermanfaat dalam hidup pribadi maupun profesi kalian saat harus mengambil keputusan yang optimal.

Pertanyaan Pemantik

Kita fokus dulu ke bidang-bidang yang akan dikaji. Perubahan penting apa yang terjadi dalam dunia pertanian dan kesehatan dalam 10 tahun terakhir? Apa yang menyebabkan perubahan tersebut?

Peta Konsep

Gambar 3.1 Peta Konsep Strategi Algoritmik dan Pemrograman

Apersepsi

Revolusi industri 4.0 menyentuh hampir semua aspek dalam kehidupan sehari-hari. Dengan pemanfaatan Internet of things (IoT), automasi, teknologi cerdas di berbagai bidang, termasuk

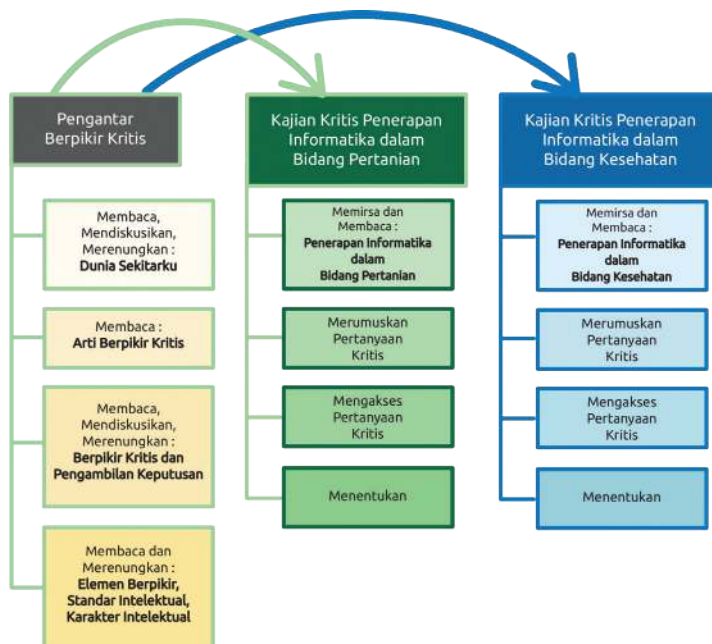
dalam bidang pertanian dan kesehatan. Sebagai contoh dalam pemanfaatan teknologi di bidang pertanian modern 4.0, petani dapat memantau kapan saatnya panen, kapan dan berapa banyak air yang harus dialirkan, tanpa harus datang ke sawah. Lalu pemasaran dan penjualan produk pun menjadi lebih efisien karena dapat dilakukan secara daring. Contoh pemanfaatan teknologi lainnya di bidang kesehatan, yaitu para dokter dapat melakukan operasi jarak jauh terhadap seorang pasien dari lokasi yang berbeda dengan lokasi tempat pasien dirawat. Terlebih pada masa pandemi di mana pertemuan langsung dibatasi dengan istilah social distancing, metode konsultasi dokter pun berevolusi, dengan hadirnya pilihan dapat dilakukan secara daring.

Perlu dipahami bahwa teknologi itu ibarat pisau bermata dua. Di satu sisi, kehadiran teknologi internet sangat memudahkan distribusi dan akses informasi yang kita butuhkan, namun di sisi lain mengakibatkan beredar dengan mudahnya berbagai macam berita bohong (hoaks). Lalu, bagaimana kita harus bersikap secara tepat terhadap berbagai macam produk TIK dan sistem komputasi tersebut? Untuk menjawab pertanyaan tersebut, kalian membutuhkan kemampuan berpikir kritis dalam mengkaji kasus-kasus sosial terkait produk TIK dan sistem komputasi yang secara lengkap akan kalian pelajari dalam Bab 3 ini agar dampak negatifnya dapat diantisipasi serta manfaatnya dapat meningkatkan kualitas hidup orang.

Kata Kunci

Berpikir kritis, pengambilan keputusan, elemen berpikir, standar intelektual, karakter intelektual, informatika dan pertanian, informatika dan kesehatan.

A. Pengantar Berpikir Kritis

Di bagian ini kalian akan belajar tentang apa itu berpikir kritis, mengapa berpikir kritis itu penting, dan bagaimana cara melatih diri untuk berpikir kritis serta mempraktikkannya pada kasus terkait Dampak Sosial Informatika di bidang pertanian dan kesehatan. Kalian akan memulainya dengan mencermati dan merefleksikan kejadian yang sering dijumpai dalam hidup sehari-hari, menyimak bacaan tentang apa itu berpikir kritis, serta keterkaitannya dengan pengambilan keputusan. Kalian juga akan berdiskusi untuk mempraktikkan berpikir kritis dalam mengambil keputusan penting yang pernah kalian alami. Selanjutnya, kalian akan menyimak video dan membaca beberapa artikel untuk mendalami arti dan hal-hal prinsip tentang berpikir kritis yang akan menjadi bekal kalian mempelajari bagian selanjutnya dalam bab ini. Gambar 3.2 berikut ini memberikan gambaran peta belajar kalian pada bab ini.

Gambar 3.2 Peta Belajar Materi Dampak Sosial Informatika

1. Dunia Sekitarku

Dalam era digital ini, berita dan informasi tersedia dalam berbagai bentuk mulai dari media cetak hingga digital yang disebarluaskan melalui berbagai cara. Beragamnya latar belakang masyarakat, menyebabkan respons orang terhadap suatu informasi pun beragam. Akibatnya, terkadang kita temukan peristiwa-peristiwa yang tidak diharapkan, bahkan merugikan atau membahayakan nyawa. Sebagai contoh, cukup sering kita mendengar atau membaca berita tentang orang yang tertipu pesan singkat yang dikirimkan melalui ponsel.

Gambar 3.3 Ilustrasi Stop Hoaks atau Penyalahgunaan Data

Di masa pandemi Covid-19, hingga bulan April 2021 Kementerian Komunikasi dan Informatika bahkan mencatat 1.733 hoaks terkait Covid-19 dan vaksin (Agustini, 2021). Beberapa berita hoaks terkait vaksin dan covid bahkan berujung pada kehilangan nyawa karena orang mempercayai berita yang tidak benar sehingga mengambil keputusan yang keliru ketika terpapar (Kurnia, 2019). Kita dapat menemukan pula berita tentang para remaja yang membahayakan nyawa dengan menghadang truk demi bisa menghasilkan konten yang disukai orang di media sosial (detikcom, 2021). Kasus lain yang juga kita jumpai adalah tentang pencurian data-data pribadi, yang terekam secara digital, untuk kejahatan (Annur, 2021).

Ayo Berdiskusi!

Aktivitas Berkelompok

Aktivitas BKDSI-K11-01-U: Dunia Sekitarku

Setelah menyimak beberapa peristiwa dalam pengantar materi di atas, selanjutnya diskusikan beberapa pertanyaan berikut dalam kelompok.

Mari Sadari!

Dalam diskusi bisa saja masing-masing anggota kelompok memiliki pendapat berbeda. Dalam kehidupan nyata, kita akan sering menghadapi perbedaan semacam ini. Justru dari perbedaan pendapat ini kalian akan bisa belajar untuk saling menghargai, mencari kesepakatan dengan menimbang secara kritis pendapat masing-masing.

Pertanyaan:

1. Mengapa berbagai peristiwa di atas terjadi?
2. Apa yang menyebabkan orang mudah mempercayai berita dan mengikutinya?
3. Apa yang harus dilakukan orang agar peristiwa semacam itu tidak terjadi?

Presentasikan hasil diskusi kelompok dalam kelas sesuai arahan dari guru. Tuliskan hasil kegiatan ini pada Buku Kerja kalian.

Mari Sadari!

Segala sesuatu yang kalian amati, pelajari, atau alami akan bermakna bagi kalian bila kalian bisa mengaitkannya dengan hidup kalian: apa hubungannya dengan pengalaman kalian sebelumnya, apa yang akan kalian lakukan selanjutnya? Memilah dan memilih: memilah yang positif dan yang negatif, kemudian memilih suatu pilihan dengan sadar untuk mencapai suatu tujuan.

Merenung dengan kritis akan menghubungkan apa yang kalian pikiran dengan apa yang kalian lakukan.

Setelah mencermati berbagai peristiwa sehari-hari di atas dan mendiskusikannya, jawablah beberapa pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja:

1. Apakah selama ini kalian mengecek terlebih dahulu kebenaran informasi yang kalian terima sebelum meneruskannya ke orang lain?
2. Apa manfaat atau kerugian yang kalian dapatkan dari jawaban kalian atas pertanyaan nomor 1 di atas?
3. Berkaitan dengan pokok pembicaran dalam bagian ini, niat pribadi apa yang muncul setelah melakukan diskusi dalam kelompok?

2. Apa itu Berpikir Kritis?

Sebelum kita membahas lebih jauh mengenai pengertian dari berpikir kritis, kalian akan melakukan aktivitas terlebih dahulu agar pemahaman yang diperoleh dapat benar-benar kalian pahami dan sesuai dengan tujuan pembelajaran pada materi ini. Mari lakukan aktivitas berikut dengan sungguh-sungguh

Ayo Membaca!

Aktivitas Individu

Aktivitas BKDSI-K11-02-U: Arti Berpikir Kritis

Di bagian ini, kalian akan belajar membaca secara kritis.

Mari Sadari!

Membaca secara kritis bukan sembarangan membaca.

Di saat membaca suatu bacaan secara kritis, kalian perlu memahami, memaknai, menginterpretasi, dan merefleksi bacaan tersebut. Jadi, kalian tidak sekedar membaca teks tanpa makna, melainkan terlibat sepenuh hati untuk mencerna hal yang dibahas, serta mengungkapkan hasil berpikir kritisnya dalam bentuk tulisan, gambar, lisan, atau bentuk lainnya.

Berikut ini bahan bacaannya.

"Literasi Digital"

Fakta menunjukkan bahwa Indonesia merupakan negara dengan jumlah pengguna internet terbanyak ke-3 di Asia (Kusnandar, 2021). Di sisi lain, menurut survei Global World Digital Competitiveness Index, literasi digital Indonesia berada di peringkat ke-56 dari 63 negara yang disurvei (IMD, 2021).

Literasi digital adalah kemampuan individu untuk menemukan, mengevaluasi, dan mengkomunikasikan informasi secara jelas dengan menulis atau menggunakan berbagai macam platform digital lainnya (Wikipedia, 2021). Literasi digital dibutuhkan untuk hidup, belajar, dan bekerja dalam masyarakat dimana komunikasi dan akses ke informasi melalui teknologi digital seperti internet, media sosial, dan perangkat bergerak (mobile devices) meningkat pesat (https://westernsydney.edu.au).

Kemampuan mengevaluasi informasi adalah salah satu kemampuan yang dibutuhkan dalam literasi digital. Untuk bisa mengevaluasi informasi, tentu saja orang butuh kemampuan berpikir yang kritis. Dalam bab ini, kalian akan mengenal apa itu berpikir kritis melalui bacaan yang disajikan. Selanjutnya kalian juga akan belajar untuk menerapkan berpikir kritis untuk menganalisis kasus dari suatu sumber bacaan.

Kemampuan berpikir adalah kemampuan yang dimiliki oleh semua manusia yang sehat. Kemampuan itu membantu untuk bertahan hidup dan mengatasi berbagai persoalan yang dihadapinya. Semakin tinggi kemampuan berpikir seseorang maka semakin berkualitas pula keputusan-keputusan yang diambil, pekerjaan yang dilakukan, dan karya yang dihasilkan.

Dapatkah kalian bayangkan, apa jadinya jika orang tidak pernah berpikir sungguh-sungguh dan mendalam ketika menghadapi sesuatu. Sebagai contoh, ketika seseorang menerima pesan singkat lewat ponsel dari nomor yang tidak dikenalnya bahwa dia mendapat hadiah mobil dari suatu perusahaan dan diminta mentransfer pajak hadiah ke nomor

rekening tertentu atau meminta nomor PIN kartu ATM untuk mengirimkan "hadiah". Karena tidak jernih berpikir, ada orang yang percaya pada pesan singkat tersebut lalu menurutinya. Akibatnya, orang tersebut tertipu dan mengalami kerugian. Keinginan untuk mendapatkan keuntungan dalam waktu singkat kadang membuat orang berpikir pendek dan tidak kritis.

Banyaknya berita hoaks yang beredar dan dipercaya cukup banyak orang, juga menjadi salah satu penanda bahwa orang sangat mungkin tidak berpikir jernih, kritis, dan mendalam ketika mendapatkan suatu informasi. Berbagai keputusan dan tindakan yang diambil tanpa disertai pertimbangan pemikiran yang jernih dan kritis bisa membawa akibat yang merugikan bagi yang bersangkutan. Oleh karena itu, kemampuan berpikir kritis penting sekali demi lebih baiknya kualitas hidup seseorang.

Sesungguhnya apa yang dimaksud dengan berpikir kritis? Dalam berbagai literatur dapat ditemukan banyak ahli yang membahas tentang berpikir kritis, misalnya Paul & Elder (2006), Glasser (2017), Clarke (2019). Masing-masing mempunyai cara pandang sendiri. Dalam buku ini, akan digunakan referensi yang ditulis oleh Paul dan Elder (2006), yang mendefinisikan berpikir kritis (*critical thinking*) sebagai seni menganalisis dan mengevaluasi pemikiran dengan maksud untuk memperbaikinya. Dengan kata lain, saat orang menghadapi suatu situasi atau mendapatkan suatu informasi baik lisan maupun tertulis, orang yang berpikir kritis akan mengambil sudut pandang tertentu kemudian menganalisis situasi atau informasi tersebut untuk mengambil kesimpulan, mengambil keputusan, atau merumuskan solusi yang tepat.

Perlu diingat bahwa berpikir kritis tidak sama dengan "asal mengkritik tanpa data dan argumentasi jelas" atau berpikir negatif atas sesuatu yang dibahas (westernsydney.

edu.au/studysmart). Sebagai contoh, ketika kalian membaca suatu artikel, kalian boleh saja setuju atau tidak setuju pada apa yang dikatakan penulis. **Jika kalian setuju**, kalian bisa tetap bersikap kritis pada isi tulisannya dengan cara menyampaikan **alasan** mengapa kalian berpikir bahwa penulis telah mengungkapkan hal yang benar dengan dasar bukti dan argumentasi yang penulis paparkan. **Jika kalian tidak setuju**, kalian bersikap kritis dengan mencoba memahami sudut pandang penulis serta bagaimana penulis bisa mengambil kesimpulan yang dimaksud, kemudian sampaikan alasan ketidaksetujuan kalian secara logis dan tetap santun. Dengan cara ini kalian akan tetap bisa menghargai orang yang berbeda pendapat dengan kalian. Sebuah botol yang isinya hanya separuh dapat disimpulkan sebagai separuh kosong, atau separuh isi.

3. Berpikir Kritis dan Pengambilan Keputusan

Dalam hidup sehari-hari, barangkali kalian juga sudah pernah menerapkan keterampilan berpikir kritis ini. Misalnya saat kalian memutuskan akan mengenakan baju apa saat menghadiri acara di sekolah, membeli tas sekolah yang seperti apa, melewati jalan yang mana ketika akan pergi ke suatu tujuan, dan sebagainya. Dalam situasi-situasi tersebut, mungkin kalian tidak asal melakukan apa yang diperintahkan oleh orang lain, namun kalian membuat keputusan berdasar beberapa pertimbangan. Perhatikan ilustrasinya pada gambar 3.4 berikut.

Gambar 3.4 Ilustrasi Pengambilan Keputusan Berdasarkan Pertimbangan

Semakin penting keputusan yang perlu diambil, semakin penting pula keterampilan berpikir kritis diterapkan. Apa yang harus dilakukan saat orang harus mengambil keputusan? Gambar 3.5 berikut ini merupakan langkah umum orang saat mengambil keputusan, mulai dari identifikasi masalah hingga evaluasi keputusan dan akibatnya. Tidak semua keputusan harus secara kaku mengikuti langkah umum ini karena bisa saja menyesuaikan kasus yang dihadapi. Di buku ini akan digunakan tahapan langkah seperti dalam Gambar 3.5 untuk membahas contoh kasus pengambilan keputusan.

Gambar 3.5 Proses Pengambilan Keputusan

Selanjutnya Tabel 3.1 berikut ini merupakan penjelasan lebih mendetail dari proses pengambilan keputusan dalam Gambar 3.5 di atas:

Tabel 3.1 Penjelasan Proses Pengambilan Keputusan

No.	Langkah	Penjelasan
1	Identifikasi Masalah	Rumuskan dengan jelas masalah yang harus diambil keputusannya, apa tujuannya, serta batasan-batasan yang ada.
2	Identifikasi Kriteria	Tetapkan kriteria yang akan dipergunakan untuk mengambil keputusan. Jika diperlukan, dapat ditetapkan sub kriteria untuk merinci setiap kriteria.
3	Tetapkan Bobot Kriteria	Tiap kriteria maupun sub kriteria bisa memiliki tingkat kepentingan yang berbeda sehingga diberikan bobot. Semakin penting kriteria/sub kriteria tersebut maka dapat diberikan bobot yang lebih besar daripada kriteria/sub kriteria lainnya.
4	Identifikasi Alternatif	Pada tahap ini diidentifikasi beberapa alternatif yang mungkin dipilih sebagai alternatif solusi, konsekuensi dari setiap alternatif, serta informasi lain terkait alternatif tersebut misalnya tentang efektivitas alternatif tersebut.
6	Evaluasi Alternatif	Evaluasilah setiap alternatif berdasar kriteria dan bobot yang ditetapkan. Gunakan informasi yang didapatkan untuk mengevaluasi.
7	Pilih Alternatif Terbaik	Pada tahap ini dapat dipilih berbagai macam metode formal untuk mendukung pengambilan keputusan seperti metode Simple Multi Attribute Rating Techniques (SMART), Analytic Hierarchy Process (AHP), Cost-Benefit Analysis, dsb (Baker, et.al., 2001). Dapat juga digunakan metode pengambilan keputusan yang dibuat sendiri berdasar kesepakatan pihak-pihak yang berkepentingan. Hasil dari tahap ini umumnya adalah urutan prioritas (ranking) alternatif untuk dipilih sebagai keputusan.

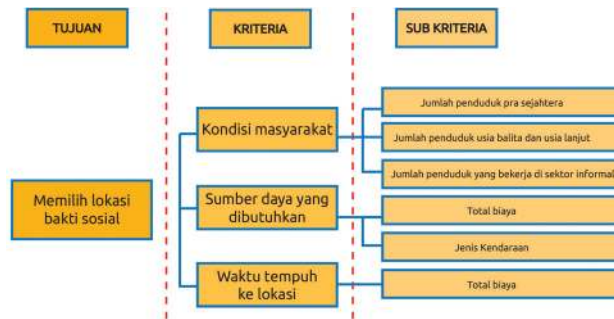
No.	Langkah	Penjelasan
8	Implementasi Keputusan	Pilih salah satu alternatif yang terbaik sesuai kriteria yang ditentukan, dari antara alternatif yang sudah dievaluasi pada tahap sebelumnya.
9	Evaluasi Keputusan	Lakukan aksi sesuai keputusan yang diambil. Dalam tahap terakhir ini, cermati hasil dari aksi yang dilakukan berdasar keputusan yang diambil. Evaluasi apakah keputusan tersebut berhasil mengatasi persoalan yang dihadapi. Jika keputusan belum berhasil mengatasi persoalan maka mungkin perlu mengulangi langkah-langkah sebelumnya sambil berpikir apa yang perlu diperbaiki untuk membuat keputusan baru. Misalnya mengumpulkan informasi lain atau informasi yang lebih rinci untuk mengeksplorasi alternatif keputusan yang berbeda atau jika perlu mengganti strategi pengambilan keputusan.

Contoh Kasus Pengambilan Keputusan

OSIS suatu SMA akan mengadakan bakti sosial (baksos) dalam rangka memperingati dies natalis sekolah. Pengurus OSIS harus memutuskan lokasi bakti sosial. Sasaran utama bakti sosial adalah balita dan warga lansia. Kegiatan bakti sosial ini harus dilaksanakan sebelum masa ujian akhir semester selama maksimal 1 hari, menggunakan dana yang telah dikumpulkan sebelumnya sebesar Rp 30 juta termasuk untuk operasional kepanitiaan. Berikut ini proses pengambilan keputusan dari pengurus OSIS dengan mengikuti proses seperti dalam Gambar 3.5.

Tabel 3.2 Contoh Proses Pengambilan Keputusan

No.	Langkah	Penjelasan
1	Identifikasi Masalah	Memilih lokasi bakti sosial dalam rangka dies natalis sekolah. • Batasan yang harus diperhatikan: • Sasaran baksos: warga miskin, balita dan lansia • Pelaksanaan: maksimal 1 hari, tanpa menginap • Dana tersedia: Rp 50 juta
2	Identifikasi Kriteria	Untuk memilih lokasi, akan dipertimbangkan 3 hal: • Kondisi masyarakat di daerah tersebut sejauh mana membutuhkan bantuan • Sumber daya yang dibutuhkan untuk melakukan bakti sosial di lokasi terkait • Waktu tempuh ke lokasi • Untuk kondisi masyarakat dan sumber daya yang dibutuhkan, dirinci lagi beberapa sub kriterianya seperti pada Gambar 3.6. Untuk waktu tempuh tidak ada sub kriteria yang lebih rinci lagi sehingga waktu tempuh menjadi kriteria sekaligus sub kriteria.
3	Tetapkan Bobot Kriteria	Setiap sub kriteria yang merupakan rincian dari kriteria diberi bobot angka (misal antara 1-5) sesuai kepentingannya. Semakin tinggi angkanya menunjukkan semakin besar bobot sub kriteria tersebut.
4	Identifikasi Alternatif	Dalam kasus ini, panitia mengumpulkan informasi dari berbagai alternatif lokasi tentang hal-hal yang terkait kriteria, mulai dari jumlah penduduk pra sejahtera hingga waktu tempuh menuju lokasi. Misal didapatkan 3 alternatif lokasi.

No.	Langkah	Penjelasan
6	Evaluasi Alternatif	Untuk setiap calon lokasi bakti sosial, panitia mengevaluasi berdasar kriteria yang ditetapkan dan menggabungkannya dengan bobot. Panitia melakukan penilaian untuk setiap calon lokasi bakti sosial dengan metode Simple Multi Attribute Rating Techniques (SMART) hingga diperoleh ranking dari setiap calon lokasi.
7	Pilih Alternatif Terbaik	Dipilih lokasi dengan *ranking* tertinggi.
8	Implementasi Keputusan	Dilakukan aksi bakti sosial di lokasi yang sesuai keputusan yang diambil.
9	Evaluasi Keputusan	Dilakukan evaluasi pelaksanaan bakti sosial di lokasi yang sudah dipilih.

Gambar 3.6 Contoh Kriteria & Sub Kriteria sebagai Faktor Penentu Keputusan

Contoh di atas adalah sebuah contoh kasus pengambilan keputusan yang sederhana dan terstruktur. Ada kalanya kasus yang dihadapi lebih kompleks. Semakin kompleks kasusnya, semakin tidak terstruktur keputusan yang harus diambil. Proses pengambilan keputusannya juga lebih rumit dan tidak seluruhnya prosedural. Dalam kasus-kasus yang kompleks, kadang banyak hal belum diketahui secara pasti sehingga terkadang intuisi dan kreativitas juga berperan penting dalam pengambilan keputusan semacam itu.

Ayo Berdiskusi!

Aktivitas Berkelompok

Aktivitas BKDSI-K11-03-U: Mengambil Keputusan Secara Kritis

Pada bagian ini kalian akan berdiskusi di dalam kelompok untuk berlatih mengambil keputusan dengan menimbang secara kritis faktor-faktor yang mempengaruhi pengambilan keputusan. Dalam kelompok, temukan sebuah kasus nyata pengambilan keputusan yang pernah atau mungkin akan kalian alami suatu saat nanti. Kalian juga bisa mengingat pengalaman saat mengerjakan tugas kelompok dalam mata pelajaran lain. Mungkin saat itu kalian harus memutuskan sesuatu terkait tugas tersebut.

Untuk kasus tersebut, temukan:

- tujuan pengambilan keputusan
- kriteria maupun sub kriteria yang akan dipakai untuk mengambil keputusan
- alasan penetapan kriteria dan sub kriteria

Gambarkan dalam bentuk bagan seperti pada contoh Gambar 3.6 Guru akan memberi kesempatan setiap kelompok untuk mempresentasikan hasil diskusi kalian.

Ayo Renungkan!

Secara pribadi, cobalah ingat pengalaman hidup kalian masing-masing hingga saat ini, lalu jawablah pertanyaan-pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja kalian.

1. Keputusan terbesar apa yang pernah kalian ambil dalam hidup kalian?
2. Bagaimana kalian mengambil keputusan?
3. Apakah ada hal yang yang kalian pertimbangkan saat mengambil keputusan?

4. Jika ada, apa sajakah pertimbangan tersebut? Mengapa hal itu menjadi pertimbangan kalian?
5. Apakah ada hal yang kalian rasa perlu diperbaiki dari cara kalian mengambil keputusan waktu itu?

4. Elemen Berpikir, Standar Intelektual, dan Karakter Intelektual

Setelah mengenal tentang berpikir kritis serta kaitannya dengan pengambilan keputusan, selanjutnya kalian akan mendalami lebih jauh bagaimana berpikir kritis ini penting untuk melakukan kajian kritis terhadap suatu bacaan dengan cara merumuskan pertanyaan berdasar bacaan. Sebelumnya, kalian perlu membaca terlebih dahulu bacaan singkat berikut ini. Perhatikan catatan "Mari Sadari" di bagian Ayo Membaca! pada bagian awal bab ini.

Ayo Membaca!

Aktivitas Individu

Aktivitas BKDSI-K11-04-U: Apa itu Elemen Berpikir, Standar Intelektual, dan Karakter Intelektual

Menurut Paul dan Elder (2006), orang yang berpikir kritis akan mampu:

- Merumuskan pertanyaan dan masalah yang penting secara jelas dan tepat karena berpikir kritis selalu dimulai dari mempertanyakan (*questioning*) dan mencari tahu lebih jauh, mendalam, dan menyeluruh atas suatu hal yang menjadi pokok bahasan;
- Mengumpulkan dan menilai informasi yang relevan, serta menafsirkannya secara efektif;
- Mengambil kesimpulan dan menemukan solusi yang masuk akal, serta menguji kesimpulan dan solusinya berdasar kriteria dan standar yang relevan;

- Terbuka terhadap pemikiran alternatif, mampu mengenali dan menilai asumsi, implikasi dan konsekuensinya;
- Berkomunikasi secara efektif dengan orang lain dalam menemukan solusi atas masalah yang kompleks.

Dalam dunia VUCA yang cepat berubah ini, keterampilan berpikir kritis perlu dilatih dan dibiasakan. Bukan hanya sekedar mampu, tetapi semakin tepat, cepat, dalam menghasilkan keputusan yang lebih baik. Kemampuan berpikir kritis tidak datang begitu saja, melainkan perlu ditumbuhkembangkan dan dibiasakan dalam keseharian agar menjadi kebiasaan baik yang bermanfaat. Bagaimanakah cara membangun kebiasaan berpikir kritis tersebut? Terlebih dahulu perlu dikenali 8 elemen berpikir seperti dalam Gambar 3.7 berikut ini:

▲ **Gambar 3.7** Elemen Berpikir

Ketika seseorang berpikir, ia melakukan beberapa hal berikut:

- Mengambil sudut pandang tertentu
- Menetapkan tujuan tertentu
- Mempersoalkan suatu hal tertentu
- Memanfaatkan informasi (dapat berupa data, fakta, observasi, maupun pengalaman)

- Melakukan interpretasi dan inferensi untuk mengambil kesimpulan dan solusi
- Menerapkan teori atau konsep yang berkaitan
- Menggunakan beberapa asumsi
- Mengidentifikasi implikasi dan konsekuensinya.

Untuk membangun kebiasaan berpikir kritis, menurut Paul dan Elder (2006), perlu diterapkan 10 standar intelektual. Salah satu cara menerapkan standar adalah dengan merumuskan pertanyaan yang berkaitan dengan standar tersebut. Tabel 3.3 berikut ini adalah standar intelektual yang dimaksud, serta contoh pertanyaan yang berkaitan dengan masing-masing standar.

Tabel 3.3 Standar Intelektual (Paul dan Elder, 2006)

No.	Standar Intelektual	Definisi	Contoh Pertanyaan terkait Standar
1	Kejelasan	Bisa dipahami maknanya	Bisakah dijelaskan lebih lanjut? Bisakah diberikan contoh? Bisakah digambarkan apa yang dimaksud?
2	Akurasi	Benar, tidak ada kesalahan	Bagaimana hal itu akan dicek? Bagaimana tahu bahwa hal itu benar? Bagaimana mengujinya?
3	Presisi	Tepat, rinci hingga level tertentu	Bisakah lebih spesifik? Bisakah lebih rinci? Bisakah lebih tepat?
4	Relevansi	Terkait dengan hal yang dibahas	Apa hubungannya dengan masalah? Apa hubungannya dengan pertanyaan? Bagaimana hal itu dapat membantu menghadapi persoalan?

No.	Standar Intelektual	Definisi	Contoh Pertanyaan terkait Standar
5	Kedalaman	Adanya keterkaitan antara banyak elemen, yang akibatnya membutuhkan kecermatan berpikir melalui banyak variabel, berbagai konteks, ide atau pertanyaan	Apa yang menyebabkan masalahnya sulit? Apa kompleksitas pertanyaannya? Apa saja kesulitan yang perlu diatasi?
6	Keluasan	Mencakup berbagai sudut pandang dan perspektif yang luas, komprehensif	Apakah perlu melihat perspektif lain? Apakah perlu mempertimbangkan sudut pandang lain? Apakah perlu melihat dengan cara lain?
7	Logika	Secara keseluruhan masuk akal, tidak ada kontradiksi;	Apakah semuanya masuk akal? Apakah awal dan akhirnya sejalan? Apakah yang disampaikan sesuai bukti nyata?

No.	Standar Intelektual	Definisi	Contoh Pertanyaan terkait Standar
8	Signifikansi	Penting, menjadi konsekuensi	Apakah hal tersebut adalah problem terpenting yang harus dipertimbangkan? Apakah hal tersebut merupakan ide pokok yang perlu dijadikan fokus perhatian? Manakah yang paling penting dari antara fakta-fakta yang ada?
9	Keadilan	Tidak memuat hal-hal berikut: bias, ketidakjujuran, pilih kasih, kepentingan diri sendiri, kebohongan, atau ketidakadilan	Apakah saya punya kepentingan pribadi terhadap masalah ini? Apakah saya merepresentasikan sudut pandang orang lain secara simpatik?
10	Kelengkapan	Mencakup semua aspek	Apakah semua informasi yang dibutuhkan sudah didapat? Apakah ada hal fakta yang masih belum terungkap? Apakah data yang terkumpul sudah mewakili keseluruhan?

Kebiasaan berpikir kritis yang dibangun perlahan dapat kalian terapkan dalam pembelajaran mata pelajaran apapun serta dalam hidup sehari-hari. Dengan berpikir kritis, kalian akan membangun karakter intelektual seperti integritas intelektual, otonomi intelektual, empati intelektual, kerendahan hati intelektual, keyakinan dalam bernalar, keberanian intelektual, ketekunan intelektual, dan pikiran yang adil. Karakter-karakter

tersebut sangat bermanfaat bagi kehidupan pribadi maupun profesi kalian di masa mendatang. Karakter intelektual tersebut akan membuat seseorang berkepribadian baik bukan hanya bagi dirinya sendiri, tetapi juga bagi orang lain dan lingkungannya. Hubungan antara elemen berpikir, standar, dan karakter intelektual dapat dirangkum dalam Gambar 3.8 Penerapan seluruh standar pada elemen-elemen berpikir akan mengembangkan karakter intelektual.

Gambar 3.8 Bagan Kaitan Elemen Berpikir, Standar, dan Karakter Intelektual

Setelah memahami apa itu berpikir kritis, kalian akan belajar menerapkan berpikir kritis melalui beberapa aktivitas membaca dan bertanya. Ada beberapa artikel yang akan kalian baca dan selanjutnya kalian akan merumuskan pertanyaan-pertanyaan kritis dengan memperhatikan elemen berpikir dan standar di atas.

Ayo Renungkan!

Setelah membaca penjelasan tentang elemen berpikir, standar intelektual dan karakter intelektual, jawablah pertanyaan-pertanyaan berikut dan tuliskan dalam Buku Kerja kalian masing-masing.

1. Hal penting apa yang kalian temukan dari bacaan tentang "Elemen Berpikir, Standar Intelektual, dan Karakter Intelektual"?
2. Bayangkanlah seseorang dengan karakter yang berlawanan dengan karakter intelektual, misalnya seseorang yang tidak punya integritas dan tidak punya empati. Akibat apa yang mungkin terjadi akibat karakter tersebut?
3. Mengapa berpikir kritis perlu dibiasakan?
4. Bagaimana cara mulai membangun kebiasaan berpikir kritis?
5. Apa manfaat berpikir kritis bagi hidup kalian di masa sekarang maupun masa mendatang?
6. Pernahkah kalian mengalami peristiwa di mana keputusan kalian tidak sepenuhnya didasarkan pada nalar pikiran namun pada intuisi? Mengapa demikian? Apakah hal itu keliru?

Intuisi adalah proses yang memberi kita kemampuan untuk mengetahui sesuatu secara langsung tanpa penalaran analitis, yang menjembatani kesenjangan antara bagian sadar dan tidak sadar dari pikiran kita, dan juga antara insting dan nalar. Menurut para ahli psikologi, dalam hidup pada dasarnya manusia membutuhkan insting dan juga nalar untuk membuat keputusan terbaik. Menyeimbangkan kedua hal itu dapat dilakukan salah satunya dengan melakukan dialog dengan diri sendiri untuk menyadari setiap keputusan yang diambil. Menulis refleksi dan jurnal adalah pembiasaan untuk berdialog dengan diri

sendiri. Harapannya, kalian akan makin bijak dalam mengambil keputusan.

B. Kajian Kritis Penerapan Informatika dalam Bidang Pertanian

Indonesia adalah negara agraris. Menurut data Badan Pusat Statistik (BPS) tahun 2020, dari 128,45 juta orang yang bekerja di Indonesia, mayoritas (29,76%) bekerja di sektor pertanian. Sebagai negara dengan jumlah penduduk terbesar ke-4 di dunia, bidang pertanian ini sangat penting untuk menjamin ketahanan pangan agar semua penduduknya mendapatkan kecukupan pangan. Bagaimana bidang informatika berperan dalam dunia pertanian? Kehadiran revolusi industri 4.0 yang mengutamakan efisiensi apakah juga menyentuh bidang pertanian? Bagaimana bentuknya? Apakah pekerja di sektor pertanian mendapatkan manfaat atau justru sebaliknya? Melalui keterampilan berpikir kritis yang telah dikenalkan dalam pertemuan sebelumnya, kalian akan belajar untuk mengkaji secara kritis penerapan informatika dalam bidang pertanian ini.

Gambar 3.9 Ilustrasi Informatika di Bidang Pertanian

Dengan bekal pemahaman berpikir kritis yang kalian pelajari dalam bagian sebelumnya, kalian akan memirsa video dan

membaca artikel tentang penerapan informatika beserta dampaknya dalam bidang pertanian. Saat membaca artikel atau memirsa video tersebut secara kritis, kalian perlu memahami, memaknai, menginterpretasi, dan merefleksikannya. Jadi, ketika disajikan suatu artikel atau video, kalian tidak sekedar menonton atau membaca tanpa makna, melainkan terlibat sepenuh hati untuk mencerna hal yang dibahas. Setelah membaca atau memirsa, tuliskan hal-hal pokok yang kalian dapatkan. Salah satu cara untuk menemukan hal-hal pokok tersebut adalah dengan menggunakan panduan 5W + 1 H, yaitu dengan menuliskan jawaban atas pertanyaan *what*, *when, where*, *who*, *why*, dan *how* berdasar artikel/video tersebut.

Berikut ini beberapa contoh pertanyaan yang bisa memberikan gambaran untuk menemukan gagasan pokok dari suatu bahan, termasuk artikel dan video. Beberapa contoh pertanyan secara khusus dikaitkan dengan topik Dampak Sosial Informatika yang menjadi pokok bahasan dalam bab ini. Tentu saja kalian bisa membuat pertanyaan lain selain contoh ini, baik dengan menggunakan acuan 5W + 1H atau yang lain. Untuk pokok bahasan yang berbeda, kalian juga bisa membuat daftar pertanyaan yang sesuai dengan pokok bahasan tersebut.

Tabel 3.4 Contoh Pertanyaan untuk Menemukan Gagasan Pokok

Kategori Pertanyaan	Contoh Pertanyaan
What **: Apa**	1. Apa yang diuraikan dalam artikel/ video? 2. Apa tujuan uraian tersebut? 3. Apa yang melatarbelakangi uraian tersebut? 4. Apa yang diharapkan dari pembaca setelah membaca artikel atau menyimak video tersebut? 5. Konsep apa yang disampaikan melalui video/ artikel tersebut? 6. Teknologi apa yang dibahas dalam artikel/video dan apa manfaatnya? 7. Apa dampak negatif teknologi yang dibahas?

Kategori Pertanyaan	Contoh Pertanyaan
Why **: Mengapa**	Mengapa hal tersebut penting untuk dibahas?
When **: Kapan**	1. Kapan video/artikel tersebut dibuat? 2. Kapan kurun waktu uraian yang dibahas terjadi?
Where **: Dimana**	Di mana uraian yang dibahas terjadi?
Who **: Siapa**	1. Siapa yang membuat video/artikel tersebut? 2. Bagi siapakah video/artikel tersebut ditujukan? 3. Siapa tokoh utama dalam artikel/video tersebut? 4. Siapa yang merasakan manfaat dari teknologi yang dibahas?
How **: Bagaimana**	Bagaimana tindak lanjut yang disarankan penulis/ pembuat video bagi orang yang membaca/ menyimak?

Selanjutnya, di dalam kelompok kalian juga akan berdiskusi untuk berlatih menerapkan konsep berpikir kritis dengan menyusun pertanyaan kritis. Kalian juga akan berlatih menerapkan keterampilan berpikir kritis dengan melakukan asesmen terhadap pertanyaan-pertanyaan yang disusun kelompok lainnya.

Ayo Bertanya!

Aktivitas Individual dan Berkelompok

Aktivitas BKDSI-K11-05-U: Mengkaji Kritis Penerapan Informatika dalam Bidang Pertanian

1. Pada aktivitas ini, kalian secara individual diminta untuk membaca sebuah artikel ilmiah berjudul "Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika" (Delima dkk., 2016) tentang pemanfaatan informatika dalam bidang pertanian. Artikel selengkapnya bisa diakses dari tautan https://journal.uii.ac.id/Snati/article/view/6236. Jika kalian kesulitan mengakses internet, maka guru akan membantu untuk mendapatkan artikel tersebut.

2. Setelah membaca artikel tersebut, tuliskan gagasan-gagasan pokok yang kalian tangkap dari artikel tersebut dalam LKPD-01 di buku ini. Kalian juga bisa menggunakan acuan 5W+1H seperti contoh pada Tabel 3.4 untuk menemukan gagasan pokok dari artikel tersebut. Untuk setiap kategori pokok bahasan, kalian boleh menuliskan lebih dari 1 hal.

Lembar Kerja Individual

LKPD-01 Hal yang Kupahami dari Bacaan Penerapan Informatika dalam Bidang Pertanian

Dari artikel yang kubaca, aku menemukan beberapa gagasan pokok berikut ini:

Tabel 3.5 Format LKPD-01 Hal yang Kupahami dari Bacaan Penerapan Informatika dalam Bidang Pertanian

Kategori	Gagasan Pokok
What	....
Why	....
When	....
Where	....
Who	....
How	....

3. Selanjutnya, berdiskusilah dalam kelompok yang terdiri atas 4-5 orang untuk menyusun minimal 10 pertanyaan kritis terkait artikel tersebut. Gunakan pengetahuan tentang berpikir kritis yang telah kalian pelajari pada bagian A bab 3 buku ini. Rubrik yang tercantum dalam LKPD-3 yang terdapat dalam kegiatan "Ayo Lakukan" di pertemuan selanjutnya dapat menjadi acuan untuk

menyusun pertanyaan yang mendalam dan untuk mengecek ulang setiap ide pertanyaan yang kalian pikirkan, apakah sudah cukup kritis atau belum.

Selain itu, kalian juga dapat membaca kembali Bab tentang Praktik Lintas Bidang dengan topik Analisis Data. Di dalam bab tersebut terdapat contoh pertanyaan kritis terkait deforestasi yang dapat menjadi inspirasi untuk membuat pertanyaan tentang artikel berikut ini.

Tuliskan pertanyaan hasil diskusi kelompok tersebut dalam LKPD-02 berikut ini. Untuk kepentingan presentasi, kelompok dapat menuliskan LKPD-02 ini dalam kertas flip besar atau media padanannya secara online. Guru akan mengarahkan kalian tentang penggunaan media ini.

Pada pertemuan selanjutnya, kalian akan mempresentasikan pertanyaan yang telah kalian susun dan juga melakukan asesmen terhadap pertanyaan-pertanyaan yang disusun oleh kelompok lain.

LKPD-02 Daftar Pertanyaan Kritis dari Artikel "Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika"

Tantangan:

Kalian juga boleh membuat pertanyaan dalam format lain yang lebih menarik dari tabel, misalnya berupa infografis.

Tabel 3.6 Format LKPD-02 Daftar Pertanyaan Kritis dari Artikel "Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika"

No.	Pertanyaan
1.	....
2.	....
3.	....

No.	Pertanyaan
4.	....
5.	....
6.	....
7.	....
8.	....
9.	....
10.	

Ayo Lakukan!

Aktivitas Berkelompok

Aktivitas BKDSI-K11-06-U: Melakukan Asesmen Pertanyaan Kritis

Pada bagian ini, kalian diminta untuk memaparkan pertanyaan yang telah kalian susun dalam kelompok pada pertemuan sebelumnya, kemudian melakukan asesmen terhadap pertanyaan yang telah disusun oleh kelompok lain. Dengan melakukan asesmen, kalian akan belajar menerapkan berpikir kritis untuk mengevaluasi hasil berpikir kritis kelompok lain.

1. Tampilkan pertanyaan-pertanyaan hasil diskusi kelompok dengan menempelkan kertas flip besar LKPD-02 di sekeliling ruangan atau dengan menggunakan media padanannya secara *online* sesuai arahan guru. Setiap kelompok memaparkan secara singkat rangkuman dari pertanyaan yang telah disusun kelompok.
2. Masing-masing kelompok secara bergiliran mencermati daftar pertanyaan dari kelompok lain dan melakukan

asesmen. Kelompok bisa melakukan aktivitas asesmen level 1 atau level 2 sesuai arahan guru. Kalian juga bisa mengomentari pertanyaan dengan menuliskan di atas kertas *post it* dan menempelkan komentar pada pertanyaan yang kalian komentari.

Aktivitas asesmen level 1 (*)

Gunakan LKPD-03 untuk menilai apakah pertanyaan yang dirumuskan oleh kelompok yang kalian nilai sudah cukup kritis. Satu lembar tabel asesmen dipergunakan untuk melakukan asesmen terhadap satu hasil kelompok lain.

Pertanyaan terbagi menjadi 2 bagian yaitu:

- Bagian A: mengases pertanyaan dengan mengacu 10 standar intelektual
- Bagian B: mengases kualitas pertanyaan

Aktivitas asesmen level 2 (**)

Selain melakukan aktivitas level 1, pada LKPD-03 tuliskan juga alasan mengapa kalian menjawab seperti itu untuk setiap kriteria. Dengan menuliskan alasan tersebut, kalian menjadi sadar betul yang kalian pikirkan. Inilah latihan berpikir logis dan kritis!!

LKPD-03 Rubrik Asesmen Pertanyaan Kritis dari Artikel "Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika"

No.	Kriteria	Nomor pertanyaan										Alasan **)
		1	2	3	4	5	6	7	8	9	10	
A. Asesmen pertanyaan dengan mengacu standar intelektual												
1	Pertanyaan berkaitan dengan standar intelektual yang mana? Tuliskan kode huruf standarnya berdasar daftar berikut ini: a. Kejelasan b. Kebenaran c. Ketepatan d. Relevansi e. Kedalaman f. Keluasan g. Logika h. Issue penting i. Keseimbangan j. Kelengkapan											
B. Kualitas pertanyaan												
2	Apakah pertanyaan dirumuskan dengan kalimat yang jelas? (Y/T)											
3	Apakah pertanyaan dirumuskan dengan struktur kalimat yang benar? (Y/T)											
4	Apakah pertanyaan relevan (berkaitan) dengan topik dalam bacaan? (Y/T)											
5	Apakah pertanyaan masuk akal? (Y/T)											

No.	Kriteria	Nomor pertanyaan										Alasan **)
B. Kualitas pertanyaan												
6	Apakah pertanyaan memuat hal penting yang terkait dengan bacaan? (Y/T)											
7	Apakah untuk dapat menjawab pertanyaan, diperlukan informasi lain di luar naskah/ artikel yang menjadi bacaan utama? (Y/T)											
8	Apakah untuk dapat menjawab pertanyaan, orang perlu berpikir lebih dahulu? (Y/T)											
9	Apakah jawaban atas pertanyaan akan menambah lengkap informasi dari bacaan? (Y/T)											
10	Apakah jawaban atas pertanyaan akan memberikan perspektif baru terhadap topik yang dibahas dalam bacaan? (Y/T)											

Tabel 3.7 Format LKPD-03 Rubrik Asesmen Pertanyaan Kritis dari Artikel "Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika "

**) Diisi oleh kelompok yang melakukan asesmen level 2

3. Tiap kelompok mempresentasikan hasil asesmennya dalam kelas dengan menyampaikan:
   - Kesimpulan umum hasil asesmen setiap kelompok yang dinilainya
   - Rincian asesmen untuk setiap butir pertanyaan
4. Guru dan peserta didik membahas hasil asesmen dan menyimpulkannya.

Ayo Renungkan!

Setelah mendalami pokok bahasan “Penerapan Informatika dalam Bidang Pertanian” melalui membaca, berdiskusi, bertanya, dan melakukan asesmen, jawablah pertanyaan-pertanyaan berikut dalam Lembar Refleksi pada Buku Kerja kalian masing-masing.

1. Secara pribadi, bagaimana sikap kalian terhadap penerapan informatika dalam bidang pertanian? Apa harapan kalian bagi Indonesia dalam hal penerapan informatika di bidang pertanian?
2. Apa yang menurut kalian perlu diperhatikan dalam mengadopsi penerapan informatika di bidang pertanian untuk mencegah timbulnya dampak negatif yang mungkin terjadi?
3. Apakah kalian merasakan manfaat berlatih berpikir kritis melalui aktivitas yang dilakukan?
4. Apakah kalian akan menerapkan kebiasaan berpikir kritis dalam pelajaran lain dan dalam hidup sehari-hari?

C. Kajian Kritis Penerapan Informatika dalam Bidang Kesehatan

Gambar 3.10 Buddy the Robot hasil penerapan Informatika dalam bidang kesehatan

Bidang kesehatan adalah bidang yang selalu menjadi perhatian manusia karena orang selalu ingin hidup lebih sehat agar mampu beraktivitas dengan leluasa. Setiap individu di manapun berada berhak untuk hidup sehat. Berbagai macam teknologi telah diciptakan untuk membantu manusia terutama yang memiliki masalah kesehatan. Seperti apakah teknologi kesehatan di era revolusi industri 4.0? Dalam bagian ini, kalian akan mengenal lebih

jauh penerapan informatika dalam bidang kesehatan, dan mengkaji kritis dampak sosialnya dalam masyarakat. Kalian akan melakukan kegiatan seperti dalam sub bab B dalam buku, mengikuti peta belajar seperti dalam Gambar 3.2.

Ayo Bertanya!

Aktivitas Individual dan Berkelompok

Aktivitas BKDSI-K11-07-U: Mengkaji Kritis Penerapan Informatika dalam Bidang Kesehatan

1. Pada aktivitas ini, kalian secara individual diminta untuk membaca 2 buah artikel tentang penerapan informatika dalam bidang kesehatan. Berikut ini adalah tautan artikel yang disarankan untuk kalian baca:
   - today.line.me/id/v2/article/Masa+Depan+Bidang+Kesehatan+Inilah+7+Robot+Medis+Super+Canggih-abd8a10c84b8a0b7c9dc4d86c6b2b93e57715ad56c609c73f8c58ef51d084838
   - litbang.kemkes.go.id/pemanfaatan-teknologi-informasi-dalam-kajian-etik/

   Jika kalian kesulitan mengakses internet, maka guru akan membantu untuk mendapatkan artikel tersebut.

2. Setelah membaca artikel tersebut, tuliskan gagasan-gagasan pokok yang kalian tangkap dari artikel tersebut dalam LKPD-04 di buku ini. Seperti halnya yang telah kalian lakukan dalam bagian B bab ini, kalian bisa menggunakan acuan 5W+1H seperti contoh pada Tabel 3.4 untuk menemukan gagasan pokok dari artikel tersebut. Untuk setiap kategori pokok bahasan, kalian boleh menuliskan lebih dari 1 hal.

LKPD-04 Hal yang Kupahami dari Bacaan Penerapan Informatika dalam Bidang Kesehatan

Dari artikel yang kubaca, aku menemukan beberapa gagasan pokok berikut ini:

Tabel 3.8 Format LKPD-04 Hal yang Kupahami dari Bacaan Penerapan Informatika dalam Bidang Kesehatan

Kategori	Gagasan Pokok
What	....
Why	....
When	....
Where	....
Who	....
How	....

3. Selanjutnya, berdiskusilah dalam kelompok yang terdiri dari 4-5 orang untuk menyusun minimal 10 pertanyaan kritis terkait artikel tersebut. Gunakan pengetahuan tentang berpikir kritis yang telah kalian pelajari bagian A bab ini.

Tuliskan pertanyaan hasil diskusi kelompok tersebut dalam LKPD-05 berikut ini. Untuk kepentingan presentasi, kelompok dapat menuliskan LKPD-05 ini dalam kertas flip besar atau media padanannya secara daring.

LKPD-05 Daftar pertanyaan kritis dari Artikel dan Berita tentang Penerapan Informatika dalam Bidang Kesehatan

Tabel 3.9 Format LKPD-05 Daftar pertanyaan kritis dari Artikel dan Berita tentang Penerapan Informatika dalam Bidang Kesehatan

No.	Pertanyaan
1.	....
2.	....
3.	....
4.	....
5.	....
6.	....
7.	....
8.	....
9.	....
10.	....

Ayo Lakukan!

Aktivitas Berkelompok

Aktivitas BKDSI-K11-08-U: Melakukan Asesmen Pertanyaan Kritis

1. Tampilkan pertanyaan-pertanyaan hasil diskusi kelompok dengan menempelkan kertas flip besar LKPD-05 di sekeliling ruangan atau dengan menggunakan media padanannya secara online sesuai arahan guru. Setiap kelompok memaparkan secara singkat rangkuman dari pertanyaan yang telah disusun kelompok.
2. Masing-masing kelompok secara bergiliran mencermati daftar pertanyaan dari kelompok lain dan melakukan asesmen. Kelompok bisa melakukan aktivitas asesmen level 1 atau level 2 sesuai arahan guru. Kalian juga bisa memanfaatkan kertas post-it untuk mengomentari pertanyaan yang kalian baca,

Aktivitas Asesmen Level 1 (*)

Gunakan LKPD-06 untuk menilai apakah pertanyaan yang dirumuskan sudah cukup kritis. Satu lembar tabel asesmen dipergunakan untuk melakukan asesmen terhadap satu hasil kelompok lain. Pertanyaan terbagi menjadi 2 bagian yaitu:

- Bagian A: mengakses pertanyaan dengan mengacu 10 standar intelektual
- Bagian B: mengakses kualitas pertanyaan

Aktivitas Asesmen Level 2 (**)

Selain melakukan aktivitas level 1, pada LKPD-06 tuliskan juga alasan mengapa kalian menjawab seperti itu untuk setiap kriteria. Dengan menuliskan alasan tersebut, kalian berlatih untuk berpikir logis dan kritis.

LKPD-06 Rubrik Asesmen Pertanyaan Kritis dari Artikel Penerapan Informatika dalam Bidang Kesehatan

No.	Kriteria	Nomor pertanyaan										Alasan **)
		1	2	3	4	5	6	7	8	9	10	
A. Asesmen pertanyaan dengan mengacu standar intelektual												
1	Pertanyaan berkaitan dengan standar intelektual yang mana? Tuliskan kode huruf standarnya berdasar daftar berikut ini: a. Kejelasan b. Kebenaran c. Ketepatan d. Relevansi e. Kedalaman f. Keluasan g. Logika h. Isu penting i. Keseimbangan j. Kelengkapan											
B. Kualitas pertanyaan												
2	Apakah pertanyaan dirumuskan dengan kalimat yang jelas? (Y/T)											
3	Apakah pertanyaan dirumuskan dengan struktur kalimat yang benar? (Y/T)											
4	Apakah pertanyaan relevan (berkaitan) dengan topik dalam bacaan? (Y/T)											
5	Apakah pertanyaan masuk akal? (Y/T)											

No.	Kriteria	Nomor pertanyaan										Alasan **)
6	Apakah pertanyaan memuat hal penting yang terkait dengan bacaan? (Y/T)											
7	Apakah untuk dapat menjawab pertanyaan, diperlukan informasi lain di luar naskah/ artikel yang menjadi bacaan utama? (Y/T)											
8	Apakah untuk dapat menjawab pertanyaan, orang perlu berpikir lebih dahulu? (Y/T)											
9	Apakah jawaban atas pertanyaan akan menambah lengkap informasi dari bacaan? (Y/T)											
10	Apakah jawaban atas pertanyaan akan memberikan perspektif baru terhadap topik yang dibahas dalam bacaan? (Y/T)											

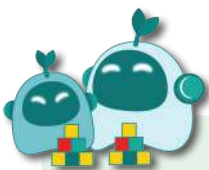
Tabel 3.10 Format LKPD-06 Rubrik Asesmen Pertanyaan Kritis dari Artikel Penerapan Informatika dalam Bidang Kesehatan

**) Diisi oleh kelompok yang melakukan asesmen level 2

3. Tiap kelompok mempresentasikan hasil asesmennya dalam kelas dengan menyampaikan:
   - Kesimpulan umum hasil asesmen setiap kelompok yang dinilainya
   - Rincian asesmen untuk setiap butir pertanyaan
4. Guru dan peserta didik membahas hasil asesmen dan menyimpulkannya.

Ayo Renungkan!

Setelah mendalami pokok bahasan "Penerapan Informatika dalam Bidang Kesehatan" melalui membaca, berdiskusi, bertanya, dan melakukan asesmen, renungkanlah pertanyaan-pertanyaan berikut dan tuliskan refleksi kalian dalam Buku Jurnal Peserta Didik.

1. Secara pribadi, bagaimana sikapmu terhadap penerapan informatika dalam bidang kesehatan? Apakah harapanmu bagi Indonesia dalam hal penerapan informatika dalam bidang kesehatan?
2. Apa yang menurutmu perlu diperhatikan dalam mengadopsi penerapan informatika dalam bidang kesehatan untuk mencegah timbulnya dampak negatif yang mungkin terjadi, termasuk yang terkait persoalan etis?
3. Setelah berlatih berpikir kritis melalui aktivitas dalam bab ini, perubahan apa yang terjadi dalam dirimu saat membaca suatu artikel?
4. Apakah kamu akan menerapkan kebiasaan berpikir kritis dalam pelajaran lain dan dalam hidup sehari-hari?

Pengayaan

Kegiatan ini dilakukan untuk meningkatkan pemahaman dan kemampuan kalian dalam materi berpikir kritis. Carilah artikel penerapan informatika dalam bidang selain pertanian dan kesehatan, yang menarik minat kalian. Bacalah dan rumuskan pertanyaan dengan cara berpikir kritis seperti dalam kasus di atas. Mintalah teman kalian untuk mengevaluasi pertanyaan yang kalian buat dengan menggunakan rubrik seperti format LKPD-03.

KEMENTERIAN PENDIDIKAN,
KEBUDAYAAN, RISET, DAN
TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas XI
Penulis : Irya Wisnubhadra
ISBN: 978-602-244-861-7

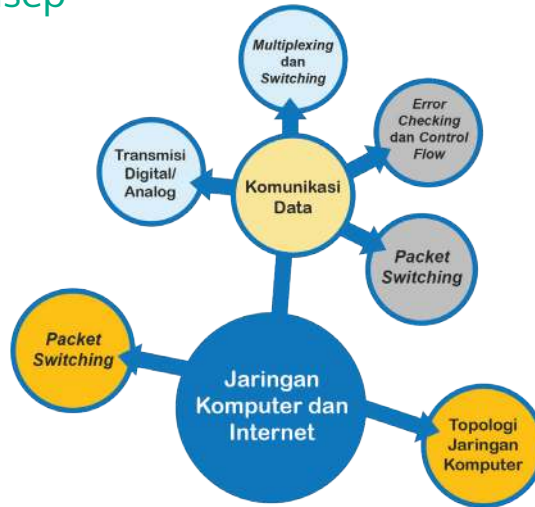
Jaringan Komputer dan Internet

Tujuan Pembelajaran

Setelah mempelajari bab ini kalian diharapkan mampu menjelaskan dan merancang topologi jaringan sederhana yang dapat menghubungkan beberapa komputer, menjelaskan dan memanfaatkan model jaringan komputer, serta mampu menjelaskan mekanisme pengiriman, penerimaan, dan komunikasi data.

Pertanyaan Pemantik

Ketika kalian berkomunikasi dan berkolaborasi dengan teman, orang tua, atau guru perlu menggunakan bahasa atau aturan. Lalu, bagaimana dengan komunikasi dan kolaborasi antar komputer?

Peta Konsep

Gambar 4.1 Peta Konsep Jaringan Komputer dan Internet

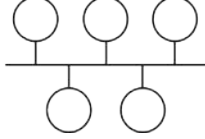
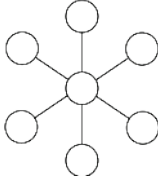
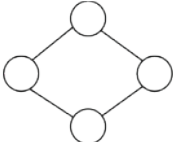
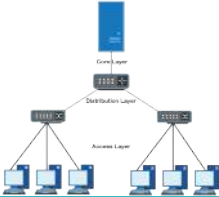
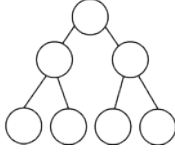
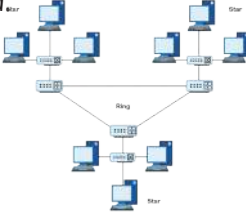
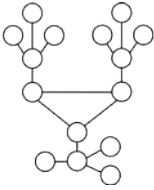
Apersepsi

Umat manusia di muka bumi terus bertambah banyak, terus berkembang hingga mencapai jumlah sekitar 7.7 miliar penduduk bumi pada tahun 2020. Umat manusia dalam kesehariannya selalu berkomunikasi satu sama lain yang pada akhirnya membentuk jaringan komunikasi antar manusia sesuai dengan wilayah, pekerjaan, hobi, dan sebagainya. Dengan jumlah umat manusia yang sangat banyak serta jaringan komunikasi antar manusia yang terus bertambah luas, dibutuhkan aturan dalam komunikasi agar dapat terjalin hubungan dengan baik dan tidak ada kesalahpahaman dalam penyampaiannya. Demikian pula yang terjadi dengan komputer pada saat ini yang terus-menerus berkembang, semakin banyak jenisnya, dan semakin bervariasi penggunaannya. Ketika antar komputer berkomunikasi dan membentuk sebuah jaringan, dibutuhkan pula aturan yang dalam hal ini disebut sebagai protokol, sehingga para pengguna komputer dapat berkomunikasi dengan efektif dan efisien.

Kata Kunci

Topologi Jaringan, Model Jaringan Komputer, Paket Data, *Error Checking and Correction*, Transmisi Analog/Digital, *Encoding*, *Modulation*.

A. Pengantar

Jaringan Komputer dan Internet adalah penemuan penting pada abad ke-20 yang mampu mengubah kehidupan manusia. Internet menjadi sumber informasi terbesar di dunia, kita sebagai pengguna internet dapat melakukan banyak kegiatan sehari-hari dengan mudah, efektif, dan efisien seperti: berbelanja, menonton film, mendengarkan musik, rapat, belajar bersama dengan guru atau teman-teman, dan masih banyak hal lainnya.

Internet terbentuk dari jaringan komputer global yang mampu menghubungkan perangkat komputer, laptop, dan gawai yang dimiliki oleh seorang individu ke jaringan atau ke piranti lainnya. Internet melibatkan miliaran perangkat yang tersambung secara global dimana satu perangkat dapat meminta layanan ke perangkat lainnya. Di sisi lain, jaringan komputer dengan skala kecil dan hanya dapat diakses secara lokal, biasa disebut dengan *Local Area Network* (LAN).

B. Topologi Jaringan

Koneksi antar komputer pada jaringan komputer baik dengan media kabel atau tanpa kabel, diatur sedemikian rupa membentuk sebuah konfigurasi jaringan. Konfigurasi jaringan dapat dipandang secara konfigurasi *logical* atau *physical*. Konfigurasi jaringan dibentuk dari topologi jaringan yang dapat direpresentasikan secara *logical* dengan bentuk lingkaran (*node*) dan garis pada *graph*. Pada konfigurasi fisik, *node* merepresentasikan komputer, *switch*, *hub*, atau *peripheral* lainnya, sedangkan garis adalah kabel penghubung atau komunikasi nirkabel.

Topologi jaringan komputer memiliki banyak jenis yang dikembangkan dengan tujuan tertentu, dengan kelebihan dan kekurangan yang dimilikinya. Jenis-jenis topologi jaringan komputer tampak pada tabel berikut:

Tabel 4.1 Jenis-jenis Topologi Jaringan Komputer dan Representasinya dalam Graph

Topologi Jaringan	Representasi Logical dalam Graph
Bus:	
Topologi bus adalah topologi dimana setiap *node* terkoneksi melalui kabel utama atau *bus* atau *backbone*. Topologi *bus* ini sangat efisien digunakan untuk jaringan komputer dengan jumlah *node* yang tidak banyak (*small network*).	
Star:	
Topologi *star* atau yang juga disebut sebagai topologi bintang adalah topologi dimana setiap *node* terkoneksi melalui *node* konektor pusat (tengah). Proses pengiriman data antar *node* dilakukan dari *node* tertentu ke pusat yang selanjutnya dikirimkan ke *node* tujuan. Topologi ini banyak dipakai karena kemudahannya	
Ring:	
Topologi *ring* adalah topologi dengan hubungan antar *node* terjadi antara satu *node* dengan *node* tetangganya dan membentuk konfigurasi lingkaran atau cincin. Pengiriman data pada topologi ini diatur dengan prinsip *token passing* (pengiriman token). *Token* berupa data kecil yang dikirimkan dari satu *node* ke *node* lainnya dalam satu arah untuk menunjukkan *node* mana yang bisa dapat mengirimkan data pada waktu tertentu.	

Topologi Jaringan	Representasi Logical dalam Graph
Tree:	
Topologi *tree* adalah topologi yang membentuk hubungan antar *node* seperti cabang pada pohon. Topologi ini biasanya digunakan untuk pengaturan jaringan pada institusi yang memiliki hirarki.	
Hybrid:	
Topologi *hybrid* adalah topologi yang menggabungkan dua atau lebih topologi jaringan, seperti topologi bintang, cincin, dan mesh. Pemilihan penggabungan topologi tergantung pada kebutuhan kinerja jaringan, banyaknya komputer yang akan dihubungkan, dan lokasinya.	

Topologi mana yang paling baik?

Tidak ada topologi jaringan paling baik untuk semua kondisi. Topologi jaringan yang tepat untuk keperluan suatu institusi bergantung pada kebutuhan dan ukuran dari jaringan yang akan dikembangkan. Hal-hal yang harus dipertimbangkan adalah:

1. Biaya Instalasi
2. Skalabilitas (kemampuan untuk perluasan jaringan dengan menambah *node*)

Biaya Instalasi

Topologi jaringan yang memiliki banyak jenis memiliki biaya instalasi yang berbeda. Semakin kompleks topologi jaringan yang akan dikembangkan semakin besar pula waktu dan biaya untuk instalasinya. Biaya dalam hal ini juga terkait dengan kebutuhan kabel, semakin jauh jangkauan jaringan semakin banyak kabel

yang dibutuhkan. Selain panjang kabel biaya juga dipengaruhi jenis kabel. Kabel serat optik (*fiber optic*) memiliki kemampuan transmisi data yang lebih cepat dan kapasitas transmisi yang lebih besar namun lebih mahal dibanding kabel tembaga (*twisted pair/coaxial*), termasuk juga komponen lain seperti *hub/switch* untuk serat optik. Dengan demikian, pemilihan topologi yang tepat harus disesuaikan dengan kebutuhan kinerja jaringan komputer, jangkauan jaringan, banyaknya komputer yang akan dihubungkan, dan biaya yang akan dikeluarkan.

Skalabilitas

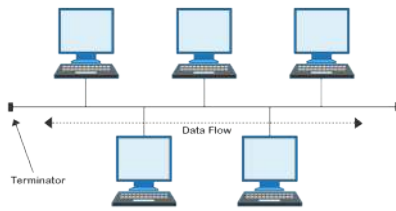
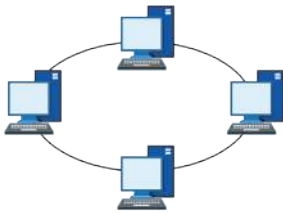
Hal lain untuk penentuan kebutuhan topologi jaringan yang tepat adalah skalabilitas. Jika institusi bermaksud mengantisipasi perluasan institusi dan jaringan komputer yang ada, maka pemilihan topologi bisa didasarkan pada kemudahan modifikasi jaringan dan penambahan *node* pada jaringan. Dengan kemudahan modifikasi, maka akan menghemat waktu dan kompleksitas instalasinya. Topologi yang mudah untuk dimodifikasi salah satunya adalah topologi bintang. Dengan topologi ini penambahan, pengurangan, dan pengubahan *node* dapat dilakukan dengan gangguan jaringan yang minimal, tidak perlu membuat *node* atau jaringan lain menjadi tidak berfungsi (*offline*). Hal ini akan sangat berbeda dengan jaringan Ring, proses penambahan *node* pada topologi ini akan menyebabkan jaringan menjadi tidak berfungsi.

Ayo Berlatih!

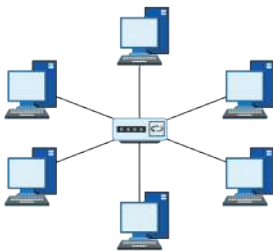
Aktivitas Kelompok

Aktivitas JKI-K11-01-U: Membandingkan dan Merancang Topologi Jaringan Sederhana

Aktivitas ini merupakan aktivitas kelompok untuk membandingkan keandalan dari beberapa topologi jaringan dan mengapa hal tersebut terjadi.

Topologi *Bus*

Pada topologi bus, tercipta dari kabel yang diberi terminator di ujung kabel. Terminator berfungsi sebagai penghenti dan penerus sinyal ke semua komputer yang terhubung pada bus. Ketika akan berkomunikasi dengan topologi ini, komputer akan menentukan alamat komputer yang akan dituju dan kemudian mengirimkan data ke komputer tujuan. Komputer lain dapat mendeteksi sinyal dari komputer yang sedang berkomunikasi.

Topologi *Ring*/Cincin

Pada topologi cincin setiap komputer terhubung dengan komputer lain menggunakan kabel yang tampak seperti gambar disamping. Setiap komputer berperan sebagai penerus pesan ke komputer berikutnya dalam satu arah.

Topologi *Star*/Bintang

Pada topologi star ada *router*/komputer yang bertindak sebagai komputer pusat. Semua komputer terkoneksi dengan router pusat. Komputer satu dapat berkomunikasi secara langsung dengan empat komputer lain melalui router pusat.

Pertanyaan:

a. Topologi mana yang paling baik jika terjadi kerusakan pada salah satu kabel jaringan, misalnya kabel terputus?

b. Bagaimana kalau ada komputer yang rusak?

c. Apa yang dapat menyebabkan kegagalan pada topologi bintang?
d. Buatlah sketsa rancangan terbaik topologi jaringan untuk 40 komputer untuk kantor dengan 4 lantai masing-masing lantai 10 komputer, dan jelaskan mengapa memilih topologi jaringan tersebut?

Ayo Berlatih!

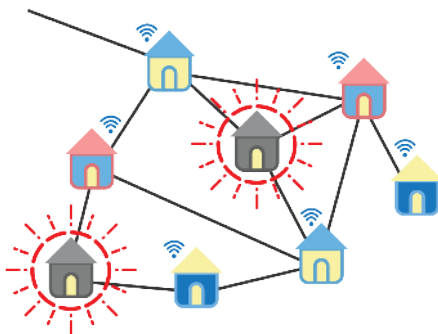
Aktivitas Individu

Aktivitas JKI-K11-02-U: Berpikir Cara Troubleshooting Jaringan Komputer dan Internet

Deskripsi:

Internet adalah jaringan komputer yang memiliki topologi jaringan tertentu yang mengarah pada *hybrid* topologi.

Kasus:

Sebuah jaringan internet dari rumah tampak seperti pada gambar berikut. Perusahaan penyedia internet memonitor terjadinya kesalahan komunikasi (*communication error*) pada dua rumah yang ditandai dengan gambar . Akibat dari kesalahan tersebut dua rumah tersebut tidak memiliki koneksi internet. Kesalahan komunikasi disebabkan oleh jaringan penghubung dari rumah ke rumah yang terputus.

Tantangan:

Konektivitas dari rumah ke rumah harus diperbaiki semuanya, namun karena terbatasnya waktu hari ini dengan prioritas

bahwa koneksi dari rumah yang tidak tersambung harus sudah tersambung dengan internet, berapa jalur yang bisa ditunda perbaikannya? Jalur yang mana? Jelaskan alasan kalian!

Sumber: (2021-HU-01 Repair the network - Bebras Hungary)

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah kalian dapat memahami topologi jaringan komputer dan mampu membedakan keunggulan dan kelemahan dari beberapa topologi jaringan komputer?
2. Jika di laboratorium komputer sekolah atau di rumah dan lingkungan rumah kalian memiliki jaringan internet, kenalilah topologi yang digunakan oleh penyedia internet kalian, dan jika ada kerusakan, bisakah kalian menemukan kira-kira dimana letak kegagalan jaringannya?
3. Apakah kalian dapat merancang topologi sederhana dari jaringan komputer?
4. Apakah materi topologi jaringan menarik buat kalian? Kesulitan apa yang kalian temukan pada materi ini?

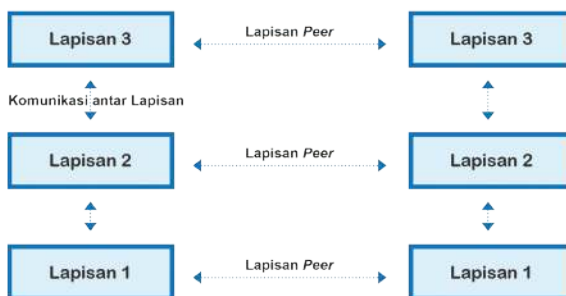
C. Model Jaringan Komputer

Rekayasa Jaringan Komputer seperti halnya konsep *engineering* pada umumnya, terdekomposisi atas banyak komponen penting yang membentuknya, yaitu: perangkat lunak, *firmware*, rekayasa chip, perangkat keras, dan sinyal listrik untuk komunikasi.

Jaringan komputer dikembangkan dengan tujuan agar sebuah komputer (*host*) dapat berkomunikasi dengan komputer lain. Ketika akan berkomunikasi, dua komputer memerlukan protokol atau aturan tertentu, seperti juga manusia menggunakan

bahasa tertentu untuk berkomunikasi. Untuk mempermudah pemahaman komunikasi antar komputer dalam jaringan jaringan yang kompleks, jaringan komputer dituliskan dalam model jaringan komputer. Model jaringan komputer berisi protokol dan aturan yang digunakan untuk komunikasi. Model ini terbagi dalam bentuk tingkat/lapisan/*layer* abstraksi yang memiliki fungsi tertentu, seperti manusia yang memiliki beberapa sistem untuk menunjang kehidupannya yaitu kerangka tubuh, sistem pernafasan, sistem saraf, otot, dan lain sebagainya.

Pada arsitektur model jaringan yang berlapis ini, suatu proses/ pekerjaan komunikasi antar komputer yang utuh dipecah dalam tugas tugas (*task*) kecil. Setiap tugas kecil akan dikerjakan oleh lapisan tertentu. Arsitektur model jaringan berlapis tampak pada Gambar 4.2. Setiap lapisan memiliki protokol dan aturan tersendiri, memiliki fungsionalitas khusus dan saling independen. Tiap tugas yang dikerjakan oleh lapisan pada komputer pengirim (*host*) juga akan dikerjakan oleh lapisan *peer* (*peer layer*) di komputer penerima (*remote*). Secara keseluruhan semua fungsionalitas jaringan komputer dilaksanakan pada lapisan-lapisan ini.

Keterkaitan antar lapisan diwujudkan dalam bentuk data yang diambil dari satu lapisan, di proses jadi *output* yang dikirimkan ke lapisan di bawahnya. Lapisan berikutnya pun melakukan hal yang sama. Pada lapisan paling bawah pada *remote* komputer proses dilakukan sebaliknya, output akan dikirim ke lapisan di atasnya.

Gambar 4.2 Arsitektur Model Jaringan Berlapis

Setiap lapisan pada model jaringan ini memiliki protokol, metode, dan prosedur yang dibutuhkan untuk mengeksekusi tugas yang diberikan.

OSI Model

Open System Interconnection (OSI) *Model* adalah model yang diciptakan oleh "International Organization for Standardization (ISO)" untuk mengatur standar protokol dan format data yang berbeda dari banyak perusahaan penyedia jaringan komputer. Model ini menyediakan kerangka yang terbuka dan standar untuk sistem komunikasi melalui jaringan komputer. Model ini menggunakan model lapisan yang terdiri atas 7 lapisan, seperti tampak pada Gambar 4.3. berikut:

Model OSI

Lapisan	Nama	Contoh Protokol
7	*Application Layer*	HTTP, FTP, DNS, SNMO, Telnet
6	*Presentation Layer*	SSL, TLS
5	*Session Layer*	NetBIOS, PPTP
4	*Transport Layer*	TCP, UDP
3	*Network Layer*	IP, ARP, ICMP, IPSec
2	*Data Link Layer*	PPP, ATM, Ethernet
1	*Physical Layer*	Ethernet, USB, Bluetooth, IEEE802.11

Gambar 4.3 Layer/Lapisan pada OSI Model

Application Layer: Lapisan ini bertanggung jawab untuk menyediakan antarmuka ke pengguna aplikasi. Protokol yang digunakan pada lapisan ini adalah protokol yang berinteraksi langsung dengan pengguna, seperti *hypertext transfer protocol* (HTTP), *file transfer protocol* (FTP), *domain name services* (DNS), dan lain sebagainya.

Presentation Layer: Lapisan ini mendefinisikan format dan mentranslasikan data yang akan ditransmisikan oleh aplikasi melalui jaringan dari pengirim ke penerima. Protokol yang digunakan pada lapisan ini adalah *Secure Socket Layer* (SSL), *Transport Layer Security* (TLS), *Remote Desktop Protocol* (RDP), dan lain sebagainya.

Session Layer: Lapisan ini mendefinisikan koneksi antar dua komputer saat dimulai, dipelihara, dan diakhiri yang disebut dengan sesi. Sebagai contoh, setelah otentikasi pengguna/kata sandi selesai, komputer jarak jauh mempertahankan sesi ini untuk sementara waktu dan tidak meminta otentikasi lagi dalam rentang waktu tertentu. Contoh protokol pada lapisan ini adalah NetBIOS (IBM), Point-to-point Tunneling Protocol (PPTP), dan lain sebagainya.

Transport Layer: Lapisan ini bertanggung jawab untuk pengiriman *end-to-end* antara pengirim dan penerima, seperti memecah data ke dalam paket-paket data, memberikan nomor urut ke paket-paket tersebut sehingga dapat disusun kembali pada komputer penerima. Selain itu, pada lapisan ini juga diciptakan *acknowledgement* sebagai penanda sukses tidaknya pengiriman data. Protokol pada lapisan ini adalah *User Datagram Protocol* (UDP) dan *Transmission Control Protocol* (TCP). Lapisan ini memiliki kemampuan transfer data yang andal dan transparan antara pengirim dan penerima. Lapisan ini juga melakukan *multiplexing*, pengendalian aliran, dan *error checking data*.

Network layer: Lapisan ini bertanggung jawab untuk pengalamatan *host* pengirim dan penerima secara unik pada jaringan. Protokol pada lapisan ini adalah *Internet Protocol* (IP) yang menciptakan *IP Address*, ARP, ICMP, dan lain sebagainya.

Data Link Layer: Lapisan ini bertanggung jawab untuk membaca dan menulis data dari dan ke jalur komunikasi, mengelompokkan data dalam *frame*, melakukan *error checking* dan *correction*, *control flow*, melakukan pengalamatan perangkat keras (MAC *Address*). Lapisan ini juga menentukan operasi perangkat jaringan seperti *hub*, *switch*, *bridge*, dan lain sebagainya pada level 2. Protokol pada lapisan ini seperti *Point-to-Point Protocol* (PPP), dan lain sebagainya.

Physical Layer: Lapisan ini mendefinisikan perangkat keras pada transmisi jaringan seperti *Network Interface Card* (NIC), *repeater*,

hub, dan lain sebagainya. Lapisan ini menentukan bagaimana cara sinkronisasi *bit*, arsitektur jaringan seperti *Ethernet*, topologi jaringan dan sistem pengkabelan. Pengiriman data dalam bentuk *bit* (biner) juga menjadi tanggung jawab lapisan ini, termasuk *encoding*. Protokol yang digunakan pada lapisan ini termasuk di dalamnya adalah: *Ethernet*, USB, *Bluetooth*, dan lain sebagainya.

Contoh dari penggunaan lapisan OSI pada pengiriman data adalah sebagai berikut:

1. Lapisan aplikasi menerima informasi yang akan dikirimkan dari pengguna, sebagai contoh pengiriman *email*.
2. Informasi akan ditranslasikan dengan menggunakan protokol *Simple Mail Transfer Protocol* (SMTP) dengan cara aman (SSL) ke lapisan presentasi.
3. Sesi pengiriman dari pengirim ke penerima akan diciptakan dan dipertahankan pada waktu tertentu melalui lapisan sesi. Informasi yang melewati tiga lapisan ini disebut dengan *Protocol Data Unit* (PDU).
4. Lapisan transport akan mentransformasi data ke bentuk lain yang disebut dengan segmen.
5. Segmen akan ditransformasikan menjadi *packet data* pada lapisan jaringan.
6. Pada lapisan *data link*, *packet data* akan diubah menjadi *frame*.
7. *Frame* selanjutnya akan mengalir ke lapisan *physical* dan diubah menjadi format *bit*, dan ditransmisikan dalam bentuk aliran *bit* (*bit stream*) arus listrik, gelombang elektromagnetik, gelombang cahaya, dan sebagainya.
8. Proses transformasi dari satu lapisan ke lapisan berikutnya dilakukan dengan penambahan *header* khusus dengan proses enkapsulasi yang dilakukan berulang yang kemudian dikirim ke penerima melalui media transmisi pada jaringan.

9. Setelah data sampai di penerima, proses yang terjadi adalah kebalikannya. *Bit stream* akan diubah menjadi informasi, *header* akan dibuang satu persatu dari lapisan terbawah ke lapisan di atasnya yang disebut proses de-enkapsulasi.

Transmission Control Protocol/Internet Protocol (TCP/IP)

TCP/IP adalah protokol dan standar komunikasi data yang digunakan pada jaringan internet. Protokol ini merupakan protokol yang paling banyak digunakan. Protokol TCP/IP dikembangkan pada awal 1980-an sebagai protokol yang menghubungkan komputer-komputer dan jaringan yang membentuk jaringan internet yang luas (WAN). TCP/IP membentuk jaringan terbuka dan independen terhadap mekanisme transport jaringan fisik yang digunakan.

TCP/IP menggunakan cara pengalamatan sederhana yang dinamakan IP *address* yang dapat menghubungkan jutaan komputer dengan sistem yang heterogen (seperti Windows, UNIX, Linux, dan lain-lain). Protokol TCP/IP merupakan kumpulan protokol (*protocol suite*) yang terdiri atas banyak protokol yang didukung perangkat lunak yang biasa disebut TCP/IP *stack*.

Keunggulan TCP/IP adalah standar protokol yang terbuka, sehingga tersedia secara bebas dan dikembangkan independen terhadap perangkat keras komputer maupun sistem operasi apapun. Karena didukung secara meluas, TCP/IP sangat ideal untuk menyatukan bermacam perangkat keras dan perangkat lunak, walaupun tidak berkomunikasi lewat internet.

Contoh dari penggunaan Cara Kerja TCP/IP adalah pengiriman data antara komputer pengirim dan penerima dalam suatu jaringan yang memiliki banyak komputer diperlukan alamat tujuan dan perantara untuk perjalanan bit stream yang akan dikirimkan. *Internet protocol* (IP) akan menjamin data

sampai ke tempat tujuan secara aman dan tepat. Saat seorang pengguna internet mengirim data ke komputer lain, protokol TCP/IP mulai diikuti. TCP membagi data menjadi paket-paket data kecil, menambahkan beberapa informasi untuk *error checking*, dan IP menambahkan label yang berisikan informasi alamat pada paket tersebut.

Paket-paket data dalam bentuk *bit stream* dikirim menuju perangkat tujuan yang dapat melalui jalur yang berbeda. Perangkat perantara seperti *router* dapat digunakan untuk mencari jalur tercepat menuju perangkat tujuan. *Router* juga digunakan untuk membantu pengaturan beban lalu lintas jaringan sehingga beban dapat terbagi rata.

Saat paket-paket data dengan TCP/IP sampai pada perangkat tujuan, alamat IP akan dibuka dari daftar pengiriman. Paket data akan disusun kembali sesuai dengan urutan jika paket data tidak *corrupted*. Saat perangkat penerima mengetahui ada data yang *corrupted*, perangkat akan meminta kiriman data ulang. Pada TCP/IP juga terdapat perangkat khusus yang disebut *gateway* yang memungkinkan berbagai tipe jaringan berkomunikasi dengan internet. Internet menjadi jaringan global.

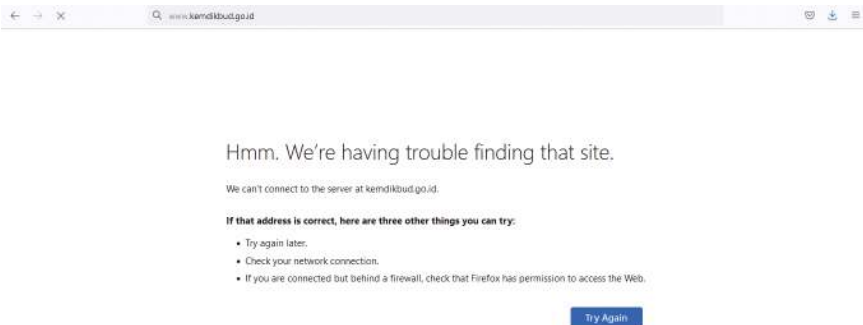
Ayo Berlatih!

Aktivitas Individu

Aktivitas JKI-K11-03-U: Pencarian Kerusakan/ Troubleshooting berdasar Model Jaringan Komputer

Deskripsi:

Suatu saat kalian sedang berselancar di laboratorium kampus di internet dengan menggunakan peramban untuk mencari informasi di situs kemdikbud.go.id.

Namun tiba-tiba koneksi internet kalian terputus, seperti tampak pada gambar berikut:

Tantangan:

Bagaimana cara agar komputer yang sedang kalian gunakan mendapatkan koneksi internet kembali yang diinspirasi dari model jaringan komputer OSI?

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah materi model jaringan komputer ini sulit bagi kalian?
2. Kesulitan apa yang kalian temukan pada materi ini?
3. Jika ada kegagalan pada jaringan internet kalian, apakah kalian dapat memanfaatkan ilmu model jaringan

komputer terjadinya untuk troubleshooting, paling tidak untuk menemukan koneksi atau komputer mana yang bermasalah?

D. Transmisi Data

Transmisi data adalah proses penting pada jaringan komputer dan internet. Ketika kita mengirimkan email atau mengakses situs *web* dan kita mendapatkan informasi maka terjadi transmisi data. Transmisi data adalah proses berbagi atau bertukar data antara dua objek, alat, atau sistem. Komunikasi data memiliki bagian penting yaitu pengirim yang melakukan proses pengiriman data, penerima yang menerima data, dan medium/kanal yang merupakan saluran transmisi dari data atau informasi dari dua objek berbeda tersebut. Bagian penting dari komunikasi data dapat dilihat pada gambar 4.4 berikut.

Gambar 4.4 Proses Pengiriman Data

Secara umum, medium atau kanal adalah benda dimana data akan bergerak dari pengirim menuju penerima. Medium ini dapat berupa udara jika data yang dikirimkan adalah data suara dari mulut ke telinga, dari radio pemancar ke pesawat radio di rumah dalam bentuk gelombang elektromagnetis, kabel tembaga, atau kabel serat optik jika data berupa cahaya yang dikirim dari satu tempat ke tempat lain. Transmisi data dari pengirim ke penerima dapat terjadi kesalahan. Kesalahan data ini dapat dideteksi dan dapat dikembalikan dengan cara tertentu.

Paket Data

Transmisi data pada jaringan komputer dan internet biasanya tidak dikirim dalam ukuran yang besar karena akan terkendala oleh kemampuan medium dan untuk kepentingan keandalan pengiriman. Data dikirimkan dalam bentuk sekumpulan paket

data yang berukuran kecil yang dinamakan data packets atau juga biasa disebut dengan datagram. Paket data dapat memiliki ukuran yang tetap atau dinamis tergantung kebutuhan sistem, namun biasanya paket data berukuran kecil, yaitu sekitar 64 KiB sehingga dapat dengan mudah dikelola.

Kumpulan paket data kemudian bisa dikirimkan melalui jalur yang berbeda-beda. Paket data ke-i, misalnya dapat melalui jalur X, namun kemudian jika suatu jalur X menjadi sangat sibuk, maka paket ke-i+1 dapat melewati jalur lain menuju tujuan dari pengiriman paket. Sama seperti ketika kurir akan mengirimkan paket belanja daring yang kita beli, tentu kurir akan mencari jalur tercepat untuk sampai ke rumah kita. Metode dengan memecah paket dalam bentuk paket-paket kecil ini sebenarnya memiliki kelemahan, yaitu dengan adanya proses penyusunan kembali paket-paket data tersebut (*reassemble*).

Struktur Paket

Paket terdiri atas tiga bagian penting, yaitu: *packet header*, *packet payload*, dan *packet trailer*. *Packet header* adalah alamat paket ketika suatu paket akan dikirimkan ke tujuan tertentu. *Packet payload* adalah paket data sebenarnya yang berisi informasi yang ditransmisikan dari sumber ke tujuan, sedangkan *packet trailer* berisi informasi penutup dari suatu paket data yang menandakan paket berakhir. *Packet trailer* dapat berisi protokol untuk pengecekan terjadinya pengiriman data. Gambar 4.3. menunjukkan contoh struktur paket data.

Gambar 4.5 Struktur Paket (*Packet Header*, *Packet Payload*, dan *Packet Trailer*)

Untuk setiap paket, *packet header*, berisi:

- Alamat IP pengirim
- Alamat IP penerima (tujuan)
- Urutan paket yang digunakan untuk penyusunan kembali paket saat sampai di tujuan
- Ukuran paket yang digunakan untuk menentukan apakah paket utuh ketika sampai di *node*, termasuk juga informasi banyaknya paket yang ditransmisikan.

Lalu untuk *Packet payload* berisi:

- Data aktual yang dikirimkan yang biasanya sebesar 64KiB.

Dan terakhir Packet trailer berisi:

- Cara untuk mengidentifikasi akhir paket
- Cara untuk mengecek kesalahan pengiriman, beberapa cara antara lain *Cyclic Redundancy Check* (CRC), checksum, dll.

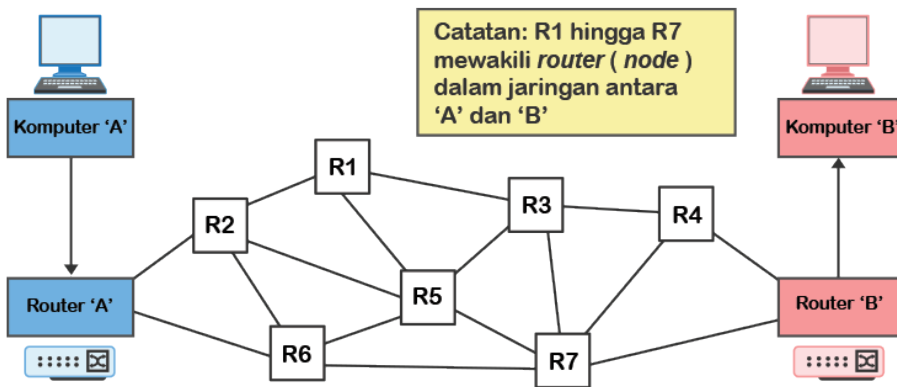
Tahukah Kalian?

KiB atau *kilobyte* adalah kuantifikasi dari informasi digital. 1 kibibyte adalah 1024 byte atau 210 byte.

1. *Packet Switching*

Packet switching adalah salah satu cara transmisi paket data yang paling banyak digunakan, dimana data akan dipecah ke dalam beberapa paket data dan dikirimkan secara independen dari titik awal ke titik tujuan. Sesampainya di tujuan paket-paket data tersebut akan disusun kembali dengan urutan yang sesuai dengan menggunakan informasi yang ada pada *header packet*. Pada setiap tahapan pengiriman data akan sampai pada *node* yang berisi *router*, yang akan melanjutkan proses pengiriman ke tujuan.

Sebagai contoh misalnya kalian sedang mengirimkan sebuah surel yang berisi foto kalian dari laptop kalian ke ponsel milik teman kalian. Surel dan foto tersebut akan dipecah dalam beberapa paket data sebelum dikirimkan.

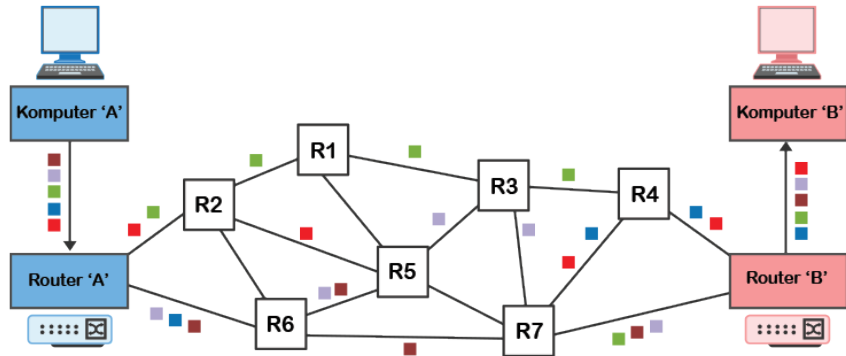
Paket-paket data akan dikirim melalui jalur atau *rute* dari komputer A ke komputer B, melalui *node* yang bisa berupa *router*, *switch*, komputer, dan lain sebagainya. Ketika sebuah paket data dikirimkan dan mencapai suatu *node* tertentu, *node* akan mengirimkan paket tersebut sesuai dengan *packet header* yang ada pada paket tersebut. Sebagai contoh pada Gambar 4.6 paket data yang dikirimkan dari komputer A akan melalui jalur yang independen untuk sampai ke komputer B.

Gambar 4.6 Jalur Pengiriman Paket Data dari Komputer A ke Komputer B

Jika paket-paket data yang akan dikirim memiliki urutan berikut:

Maka paket-paket data tersebut akan mengikuti alur sebagai berikut:

1. Paket data akan ditransmisikan melewati jalur yang independen
2. Router akan menentukan kemana paket akan dikirim
3. Penentuan jalur (routing) akan ditentukan dari banyaknya paket yang sedang diproses oleh node tertentu
4. Jalur yang dipilih sebisa mungkin adalah jalur terpendek, namun jika jalur terpendek tersebut tidak tersedia atau sedang sibuk, maka jalur lain bisa dipilih.

5. Karena berbeda jalur makan paket-paket data bisa sampai di tujuan dengan urutan yang berbeda.

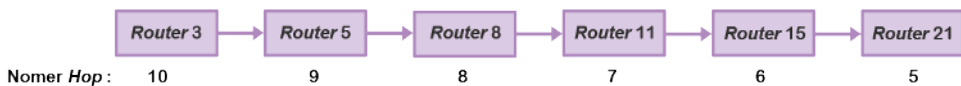
Gambar 4.7 berikut menggambarkan urutan pergerakan paket-paket data:

Gambar 4.7 Ilustrasi Pergerakan Pengiriman Paket Data dari Komputer A ke Komputer B

Metode *packet switching* dianggap sebagai salah satu cara yang andal, namun cara ini juga memiliki kelebihan dan kekurangan. Beberapa kelebihan cara ini adalah:

a. Paket data dapat dikirimkan dari sumber ke tujuan tanpa jalur khusus (*dedicated channel*)
b. Kegagalan pengiriman rendah, karena *router* dapat melakukan pengiriman ulang (*rerouting*) paket jika terjadi kegagalan.
c. Banyak user bisa menggunakan jaringan yang sama.
d. Memungkinkan pengiriman data kecepatan tinggi.

Namun *packet switching* juga memiliki beberapa kekurangan, di antaranya adalah:

a. Paket data bisa hilang dan harus dikirimkan kembali.
b. Proses menyusun kembali paket data yang membutuhkan waktu.
c. Adanya *delay* (jeda) dan rentan terhadap *error* sehingga sulit digunakan untuk pengiriman data yang cepat dan berkualitas tinggi seperti *voice call*.

Terkadang paket data pada *packet switching* hilang di perjalanan karena *bouncing data* (data yang berpindah-pindah) di antara *router* dan akhirnya tidak sampai di tujuan atau penerima data. Proses pengiriman akan berhenti yang dapat menyebabkan jaringan terhenti karena meningkatnya paket yang hilang. Untuk mengatasi hal itu, muncul teknik yang disebut dengan metode *hopping*. *Hop number* (penghitung hop) atau *Time to Leave* (TTL) akan diberikan pada *header* pada setiap paket data, dan penghitung ini akan dikurangi satu setiap kali paket data meninggalkan *router*. Awalnya paket data diberi penghitung *hop* maksimum. Jika penghitung *hop* sudah nol dan paket data tidak sampai pada tujuannya maka paket data tersebut dihapus dari *router* berikutnya. Paket yang hilang ditandai oleh komputer penerima dan permintaan untuk pengiriman ulang paket data diciptakan. Ilustrasi mengenai hopping tampak pada Gambar 4.8.

Gambar 4.8 Ilustrasi Hopping

Tahukah Kalian?

Metode lain dalam pengiriman data selain *packet switching* adalah *circuit switching* dimana jalur komunikasi dibentuk secara khusus (*dedicated*) sebelum komunikasi antar dua *node* berlangsung. Hal ini sangat berbeda dengan *packet switching* dimana tidak ada jalur khusus antara dua *node*. *Circuit switching* digunakan pada jaringan telepon analog.

Ayo Kerjakan!

Aktivitas Kelompok

Aktivitas JKI-K11-04-U: Mengurutkan Mekanisme Pengiriman Data dengan Packet Switching

Deskripsi:

Aktivitas ini adalah aktivitas untuk menentukan urutan dari *packet switching* dalam proses pengiriman data dari komputer pengirim ke komputer tujuan.

Apa yang kalian butuhkan?

1. Kertas HVS, dan printer
2. Gunting

Apa yang kalian lakukan?

1. Atur kelompok peserta didik yang terdiri dari 2 – 3 peserta didik
2. Cetak dokumen berbentuk tabel berikut pada kertas HVS
3. Potong cetakan pada kertas HVS untuk tiap kotak pada tabel.

Berikut ini langkah-langkah dari aktivitas kelompok berkaitan materi packet switching:

1.	Paket yang dikirim ke jaringan, berpindah dari *router* ke *router* dengan jalur berbeda (ditetapkan oleh *router*). Oleh karena itu, waktu perjalanan setiap paket dapat berbeda.
2.	Data dipecah menjadi beberapa *chunk* (paket)
3.	Setelah paket tiba, paket-paket tersebut disusun ulang membentuk data awal.
4.	Jika data membutuhkan banyak paket maka urutan setiap paket dicatat.
5.	Jika tidak ada pesan konfirmasi, pengirim mengirimkan data lagi.

6.	Setiap paket memiliki alamat dari, ke alamat dan *payload* (potongan data).
7.	Pesan konfirmasi terkirim dari penerima ke pengirim yang menunjukkan bahwa pesan tersebut telah diterima.

Tantangan:

Kalian diminta untuk mengurutkan langkah-langkah pengiriman data tersebut dengan metode packet switching, agar langkah-langkah yang tersedia menjadi urutan yang masuk akal.

Ayo Kerjakan!

Aktivitas Individu

Aktivitas JKI-K11-05-U: Gangguan pada Packet Switching

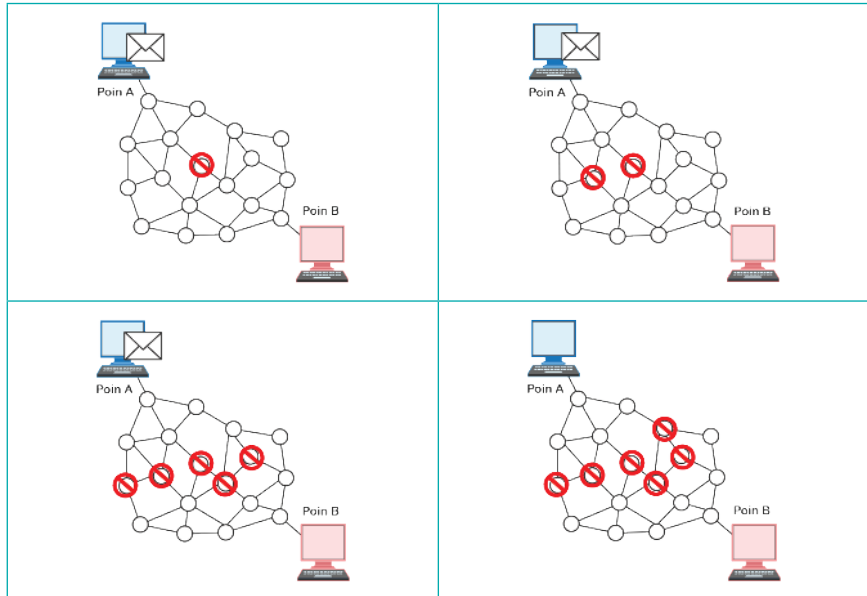
Deskripsi:

Aktivitas ini adalah aktivitas untuk memberikan pemahaman pada kalian, apa yang terjadi ketika gangguan pada node pada proses packet switching. Diketahui proses pengiriman data menggunakan packet switching sebagai berikut:

Kondisi Awal: Sebuah surel akan dikirimkan dari komputer A ke komputer B.	**Kondisi Akhir**: Surel terkirim ke komputer B dengan melewati node berwarna merah.

Tantangan:

Bagaimana jika jaringan memiliki sejumlah *node* rusak seperti beberapa kasus pada gambar berikut ini? Tugas kalian adalah menentukan *node* pengiriman data yang dilalui dari komputer

A ke komputer B dengan mewarnai *node* pada jaringan tersebut!

2. Deteksi dan Perbaikan Kesalahan

Kesalahan dapat terjadi pada pengiriman data pada jaringan komputer yang menyebabkan data menjadi termodifikasi, hilang, atau *corrupted*. Dalam kegiatan sehari-hari pun tentu kalian dapat melihat berbagai contoh kesalahan, seperti paket pesanan barang yang salah kirim alamat dan banyak hal lainnya. Selanjutnya sebagai ilustrasi dan gambaran mengenai materi deteksi dan perbaikan kesalahan, silahkan kalian baca dan perhatikan teks berikut ini.

Bacalah Teks berikut:

Apakah adna dpaat mebmaca teks ini?

Syaa tdak preacya syaa daapt mebmaca teks ini denagn mengugnakan kemampaun pikrian oatk maunsia. Tdiak menajdi maaslah utruan hruuf pdaa ktaa ynag diutlis, ktia maish dpaat memabca dngan baik.

Hal ini dikanarekan oatk manuisa membcaa ktaa secraa utuh tiadk dengan setaip huurfnya.

Menakjbukan buakn?

Bagaimana, apakah kalian dapat membaca contoh teks tersebut? Contoh teks diatas mungkin dapat dibaca dan dimengerti oleh sebagian besar orang, namun ternyata tidak demikian dengan komputer. Data yang berubah atau terkorupsi (*corrupted*) sering ditemukan pada pengiriman data seperti pada contoh teks di atas. Oleh sebab itu pengecekan kesalahan sangat penting di teknologi informatika. Adapun bentuk-bentuk kesalahan yang dapat terjadi pada pengiriman data, yaitu:

1. *Single Bit Error*

2. *Multiple Bit Error*

3. *Burst Error*

Kesalahan yang terjadi pada proses pengiriman data tersebut dapat dikendalikan dengan dua cara, yaitu:

a. Pendeteksian Kesalahan

Pendeteksian kesalahan merupakan salah satu cara yang ampuh dalam mengendalikan kesalahan ketika proses pengiriman paket data. Ada beberapa cara untuk pendeteksian kesalahan pada pengiriman data, diantaranya yaitu:

1. *Parity Check*
2. *Checksum*
3. CRC (*Cyclic Redundancy Check*) *diberikan dalam bagian Pengayaan

1) Parity Check

Parity check adalah salah satu metode untuk mengecek apakah data yang ditransmisikan sesuai dengan aslinya.

Metode ini menggunakan perhitungan banyaknya bit 1 pada data yang dikirimkan. *Parity check* dapat menggunakan *parity* genap atau gasal. *Parity* genap menggunakan aturan banyaknya bit 1 harus genap sedangkan *parity* ganjil menggunakan aturan banyaknya bit 1 pada data harus ganjil. *Parity check* diletakkan sebagai tambahan bit pada data yang akan dikirim, yang biasanya di bit paling kiri (*most significant bit*), yang biasa disebut dengan *parity bit*. Sebelum proses pengiriman data, pengirim dan penerima data harus sepakat jenis *parity* mana yang digunakan. Hal ini disebut dengan protokol transmisi.

Perhatikan contoh berikut ini!

	1	1	0	0	1	1	0

parity bit

Bit paling kiri digunakan sebagai *parity* bit. Jika metode *parity* genap yang digunakan maka *parity bit* diisi dengan bit 0. Banyaknya bit 1 adalah 4 (genap):

0	1	1	0	1	1	0	0

parity bit

Jika metode *parity* ganjil yang digunakan maka *parity bit* akan diisi dengan bit 1. Banyaknya bit 1 adalah 5 (ganjil):

1	1	1	0	1	1	0	0

parity bit

Parity check berguna ketika kesalahan pengiriman data hanya muncul pada satu bit, namun ketika kesalahan terjadi pada dua bit data, maka *parity check* tidak dapat mendeteksinya. Sebagai contoh jika data berubah pada dua bit ke 5 dan 6, maka banyaknya bit 1 masih genap dan data dianggap tidak salah. Data dikirimkan menggunakan *parity* genap dan berubah ketika diterima, namun masih memenuhi *parity* genap.

Data dikirimkan

1	1	0	1	1	0	1	1

Data diterima

1	1	0	1	0	1	1	1

Gambar 4.9 Ilustrasi *Parity Check* Kesalahan 2 bit Data

Pengembangan dari *parity check* selanjutnya adalah *checksum*. Checksum akan dibahas pada sub bab berikutnya. Pendeteksian kesalahan pada parity check masih memiliki kekurangan dimana metode ini belum dapat menemukan dimana tepatnya atau bit ke berapa dari data yang corrupted. Metode ini selanjutnya dikembangkan dengan menggunakan parity block atau memberikan tambahan parity byte.

Contoh pada tabel berikut, data sebanyak 9 byte dikirimkan dari pengirim ke penerima. Pengirim dan penerima bersepakat menggunakan parity genap. Kesepakatan juga dilakukan untuk mengirimkan parity byte yang berfungsi sebagai sebagai parity bit vertikal yang dapat digunakan untuk memastikan bit mana yang salah.

Tabel 4.2 Contoh Pengecekan *Data Corrupted* menggunakan *Parity Byte* Genap

	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7	Bit-8
Byte-1	0	1	1	0	0	1	1	0
Byte-2	0	1	0	0	1	1	0	1
Byte-3	1	0	1	1	1	0	1	1
Byte-4	0	0	1	0	1	0	1	1
Byte-5	0	1	1	1	0	0	1	0
Byte-6	0	1	0	1	0	1	1	0
Byte-7	1	0	1	1	1	0	0	0
Byte-8	1	1	1	1	1	1	1	0
Byte-9	1	0	0	1	0	0	0	0
Parity Byte	0	1	0	0	0	0	0	1

Tabel 4.2 menunjukkan contoh bit-5 dari byte-8 yang *corrupted* atau mengalami kerusakan data. Pendeteksian kesalahan pengiriman tersebut dapat diperbaiki dengan mengganti bit-5 dari byte-8 menjadi 0, sehingga byte 8 menjadi:

Byte-8	1	1	1	1	0	1	1	0

2) Checksum

Checksum adalah metode yang digunakan untuk mengecek apakah data yang dikirimkan pada saat transmisi berubah atau *corrupted*. *Checksum* ditambahkan pada blok data pada bagian akhir blok. Proses dari *checksum* adalah sebagai berikut:

1. Pada saat blok data akan ditransmisikan, *checksum* akan dihitung dari blok data
2. Perhitungan *checksum* menggunakan algoritma yang disepakati antara pengirim dan penerima data
3. *Checksum* kemudian ditransmisikan bersama dengan blok data
4. Pada penerima, *checksum* akan dihitung kembali dan dibandingkan dengan *checksum* yang ada di blok data
5. Jika hasil perhitungan ulang *checksum* sama dengan *checksum* yang ada di blok data maka pengiriman terjadi dengan tanpa kesalahan, jika berbeda maka permintaan pengiriman data ulang akan dikirimkan.

Contoh penggunaan *checksum* pada data dari pengirim, dapat dilihat pada tabel 4.3 dengan keterangan Byte-1 dan Byte-2 adalah data yang akan dikirim kepada penerima.

Data dari Sisi Pengirim:

Tabel 4.3 Contoh Penggunaan *Checksum* pada Data dari Sisi Pengirim

	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7	Bit-8
Byte-1	1	0	1	0	0	1	1	0
Byte-2	0	0	1	0	1	1	0	1
Sum(*)	1	1	0	1	0	0	1	1
Checksum (**)	0	0	1	0	1	1	0	0

*: dengan komplemen 1

**: komplemen dari Sum

Data yang diterima pada sisi penerima:

Data

Checksum

Perhatikan tabel di bawah ini, berikut ini merupakan ilustrasi pendeteksian kesalahan menggunakan checksum:

Tabel 4.4 Ilustrasi Penggunaan *Checksum* untuk mengecek Data Berubah atau *Corrupted*

	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7	Bit-8
Byte-1	1	0	1	0	0	1	1	0
Byte-2	0	0	1	0	1	1	0	1
Checksum	0	0	1	0	1	1	0	0
Sum	1	1	1	1	1	1	1	1
Komplemen Sum	0	0	0	0	0	0	0	0

Kesimpulan data *error* atau berubah atau *corrupted* dapat dilihat pada baris "Komplemen Sum", yaitu:

- Jika Komplemen Sum memiliki nilai sama dengan 0 (nol), maka tidak ada *error* pada proses pengiriman data.
- Jika Komplemen Sum memiliki nilai tidak sama dengan 0 (nol), maka terdapat error pada proses pengiriman data.

b. Perbaikan Kesalahan

Alternatif lainnya dalam pengendalian kesalahan pengiriman data yaitu dengan cara melakukan perbaikan kesalahan.

Adapun metode perbaikan kesalahan ini dapat dilakukan dengan dua cara, yaitu:

3) ***Backward Error Correction***

Backward Error Correction adalah cara perbaikan kesalahan dengan cara permintaan kembali pengiriman data dari pengirim ketika data dideteksi adanya kesalahan. Cara ini adalah cara yang mudah dan efisien jika pengiriman data mudah dilakukan, misalnya jika koneksi menggunakan serat optik. Jika pengiriman data menggunakan perangkat nirkabel maka cara ini tidak efektif.

4) ***Forward Error Correction***

Forward Error Correction adalah cara perbaikan kesalahan dengan menggunakan algoritma perbaikan kesalahan untuk mengembalikan data yang *corrupted*. Untuk dapat melakukan perbaikan kesalahan, penerima data harus mengetahui secara persis bit mana yang *corrupted* dengan menggunakan *parity bit* sebagai salah satu cara.

Ayo Kerjakan!

Aktivitas Individu

Aktivitas JKI-K11-06-U: Pendeteksi dan Perbaikan Kesalahan

1. Proses pembelajaran secara daring/*online* sedang terjadi di sekolah. Guru sedang menjelaskan materi melalui aplikasi *teleconference*. Packet *switching* digunakan dalam pengiriman suara dan video, yang dapat digambarkan sebagai berikut:

Pada saat telekonferensi sedang berlangsung ternyata video dan suara guru menjadi putus-putus dan tidak

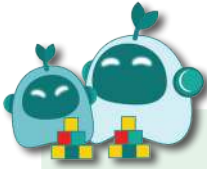
terlihat dengan baik. Coba kalian jelaskan, mengapa hal tersebut dapat terjadi?

2. Saat membuka sebuah halaman web dengan menggunakan browser, kalian akan melihat informasi (teks, gambar, atau lainnya) ditampilkan, sebagian demi sebagian. Cobalah menampilkan sebuah halaman *web* dengan *browser*, jelaskan bagaimana browser menampilkan halaman web dengan *packet switching*?

3. 2-byte data akan dikirim ke penerima dengan *checksum*. Tentukan nilai checksum dari data tersebut.

	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7	Bit-8
Byte-1	1	0	1	1	0	1	0	1
Byte-2	0	0	0	1	1	1	0	1
Sum								
Checksum								

4. Paket data dengan menggunakan *checksum* diterima dengan bit berikut:

Coba kalian perhatikan dengan seksama. Apakah data tersebut *corrupted*? Jelaskan jawabanmu!

Pengayaan

Untuk meningkatkan pemahaman dan kemampuan kalian dalam materi *Error Detection* (pendeteksi kesalahan) dari data yang dikirim. Kalian dapat mengikuti aktivitas pengayaan ini, dengan menggunakan metode yang ketiga, yaitu CRC. Silakan ikuti langkah-langkah pengayaan berikut ini:

1. Carilah penjelasan mengenai metode *Cyclic Redundancy Checks* (CRC) dari sumber terpercaya yang tersedia di internet!
2. Tentukan CRC code tambahan jika 2 *byte* data berikut akan dikirim oleh komputer A kepada komputer B, berikut rangkaian data *byte* datanya:

	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7	Bit-8
Byte-1	1	0	1	1	0	0	1	0
Byte-2	0	0	1	1	1	0	1	1
CRC Code								

3. Kalian dapat berdiskusi membandingkan jawaban kalian dengan teman atau guru, untuk mengetahui apakah CRC Code kalian sudah tepat, selamat mengerjakan!

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi dalam Buku Kerja kalian, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah kalian telah dapat memahami cara pengiriman data pada jaringan komputer dan internet?

2. Apakah kalian telah dapat memahami cara pendeteksian kesalahan pengiriman data dan melakukan perbaikan data yang *corrupted* pada jaringan komputer dan internet?
3. Dengan pemahaman kalian saat ini, apakah kalian dapat memahami mengapa ketika kalian sedang melakukan video conferencing atau sedang menelepon melalui aplikasi, suara atau gambarnya bisa terputus-putus?
4. Apakah materi topologi jaringan menarik buat kalian? Kesulitan apa yang kalian temukan pada materi ini?

3. Transmisi Digital dan Analog

Data yang berupa data teks, numerik, gambar, video, dan audio yang telah kalian pelajari pada materi-materi sebelumnya dapat dikirimkan dari satu piranti ke piranti lainnya. Data tersebut dapat dikirim melalui medium/kanal dalam bentuk sinyal elektromagnetis. Data sendiri bisa berjenis analog seperti suara manusia, gerak objek di permukaan bumi, dan berjenis digital seperti *file* pada *hard disk*, atau *flash drive* kalian. Data analog dan digital direpresentasikan dalam bentuk sinyal digital ataupun sinyal analog ketika akan ditransmisikan.

Sinyal Analog adalah sinyal yang bersifat gelombang kontinu (*continuous*) yang direpresentasikan dengan gelombang kontinu elektromagnetik untuk sinyal analog elektris. Gerak mekanik, pneumatis, dll. juga bisa disebut dengan sinyal analog.

Sinyal Digital adalah sinyal yang bersifat diskrit dan dapat digambarkan seperti tangga naik turun. Sinyal digital digunakan pada komputer, sinyal televisi digital, sinyal satelit, dll. Sinyal digital memiliki sifat yang tahan terhadap gangguan (*noise*).

Apa kalian pernah mendengar bahwa sejak tahun 2021 siaran televisi di Indonesia telah beralih menjadi siaran televisi

digital? Ya, sejak tahun 2012, melalui Peraturan Menteri Kominfo No. 05 tahun 2012, Indonesia telah mengadopsi standar penyiaran televisi digital.

Sistem penyiaran televisi digital menggunakan transmisi sinyal digital sehingga mampu memberikan layanan transmisi data gambar dan suara dengan baik, tidak menurun kualitasnya dan bahkan memiliki kemampuan tambahan seperti layanan interaktif, layanan informasi acara yang akan berlangsung dan bahkan informasi tambahan mengenai peringatan dini bencana yang bisa dikirimkan. Televisi yang dapat menangkap sinyal digital telah banyak dipasaran. Televisi analog pun masih dapat digunakan dengan menambahkan *set top box* (STB) khusus.

Ketika sinyal terkirim, sinyal sering kali menurun kualitasnya. Penurunan kualitas disebabkan oleh beberapa hal, seperti:

1. *Delay distortion* (distorsi kelambatan): yaitu penurunan kualitas yang disebabkan oleh keterlambatan/terlalu cepat nya data terkirim. Hal ini bisa terjadi karena ketidak-cocokan kecepatan dan frekuensi dari penerima dan pengirim. Pada pengiriman sinyal digital hal ini menjadikan bit bit data yang terkirim bisa terlambat atau terlalu cepat yang menyebabkan data menjadi salah.
2. *Attenuation* (atenuasi): yaitu penurunan kualitas sinyal yang bisa berasal dari amplitudonya, atau arus listrik yang berkurang, dll. Atenuasi biasanya terjadi karena adanya kehilangan kekuatannya karena jalur yang panjang. Penerimaan sinyal di sisi penerima diharapkan masih memiliki sinyal yang memadai sehingga dapat diinterpretasikan secara akurat.
3. *Noise* (derau): yaitu gangguan yang terjadi secara acak terhadap sinyal analog atau digital yang dapat menyebabkan

berkurangnya kualitas sinyal. Terdapat beberapa jenis derau, yaitu:

a. *Thermal Noise*: Derau karena suhu yang mempengaruhi sifat konduktif dari medium pengiriman
b. *Intermodulation*: Derau yang terjadi karena penggunaan medium pengiriman secara bersama oleh banyak frekuensi.
c. *Crosstalk*: Derau yang terjadi karena masuknya sinyal asing ke medium pengiriman data.
d. *Impulse*: Derau yang terjadi karena gangguan insidental seperti petir, tegangan listrik yang tidak biasa, dan kegagalan komponen. Sinyal digital adalah sinyal yang rentan terhadap gangguan derau ini

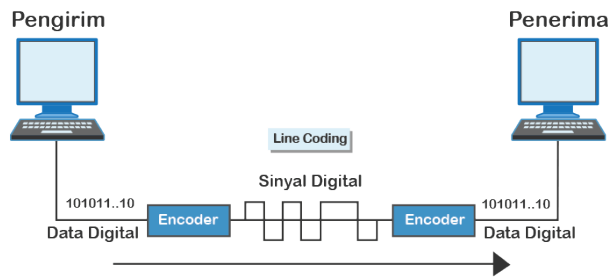
Pada pengiriman data antar komputer diperlukan medium/media transmisi yang dapat berbentuk *guided media* (media terbimbing), atau *unguided media* (media tak terbimbing). Guided media diantaranya adalah kabel *unshielded twisted pair* (UTP), kabel *coaxial*, *power line*, dan serat optik dimana pengirim dan penerima terkoneksi secara langsung. *Power line* adalah teknologi yang mengirimkan sinyal menggunakan kabel yang berisi tenaga listrik, Sedangkan *unguided media* adalah udara dimana koneksi antara pengirim dan penerima akan dilakukan secara *wireless* (tanpa kabel).

Pengiriman informasi melalui media pengiriman memiliki kapasitas yang disebut dengan *channel capacity* (kapasitas kanal). Kapasitas kanal biasa disebut dengan *data rate* yang diukur berdasarkan: *bandwidth*, *error-rate*, dan *encoding*. *Bandwidth* adalah kapasitas fisik dari media pengiriman, misalnya kabel UTP CAT7 mampu menampung kecepatan 10 Gbps, dengan panjang maksimum kabel 100 meter, serat optik mampu menampung kecepatan 200 Gbps, dengan panjang maksimum kabel 80 km.

Error-rate adalah frekuensi kesalahan yang terjadi pada pengiriman data yang disebabkan karena derau, sedangkan *encoding* adalah tingkat metode yang digunakan pada persinyalan.

a. Transmisi Digital

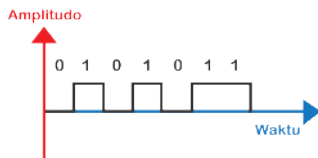
Transmisi Digital adalah proses pengiriman data secara digital. Data atau informasi bisa berbentuk analog atau digital, sedangkan data yang ada pada komputer adalah data yang berbentuk diskrit digital. Untuk mengirimkan data secara digital maka data harus diubah atau dikonversi dalam bentuk digital.

Ada dua cara transmisi data digital yaitu dengan cara *Line Coding* dan *Block Coding*. *Line Coding* adalah transmisi data dengan mengkonversi data digital menjadi sinyal digital. Data digital adalah data dalam bentuk biner 0 dan 1 yang membentuk deretan (urutan). Sinyal digital akan merepresentasikan data digital dalam bentuk data diskret, seperti tampak pada gambar 4.10.

Gambar 4.10 Pengiriman data digital dengan *Line Coding*

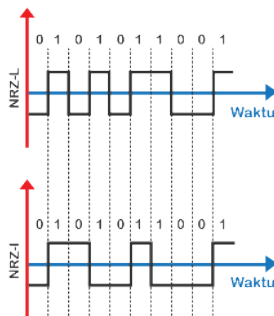
1) *Line Coding*

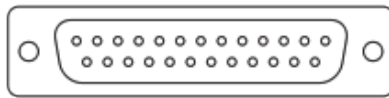
Line Coding memiliki tiga cara untuk merepresentasikan data digital menjadi sinyal digital, yaitu dengan cara *encoding unipolar*, polar, atau bipolar.

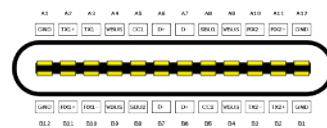
a) *Unipolar Encoding*

Unipolar Encoding adalah cara pengkodean sinyal dengan merepresentasikan data dengan menggunakan level tegangan tunggal (*single voltage level*). Bilangan biner 1 akan direpresentasikan dengan tegangan tinggi, dan bilangan biner 0 direpresentasikan dengan tanpa tegangan. *Unipolar encoding* juga disebut dengan *Unipolar-Non-Return-To-Zero*. Gambar 4.11 berikut ini menunjukkan pengkodean sinyal *unipolar*:

Gambar 4.11 *Encoding Unipolar*

b) *Polar Encoding*

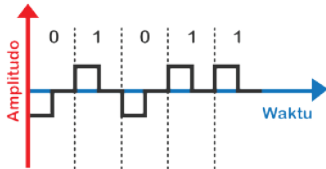
Polar encoding adalah pengkodean sinyal menggunakan tingkat tegangan yang jamak untuk merepresentasikan bilangan biner, yang memiliki 4 tipe, yaitu: *Polar Non Return to zero* (NRZ), *Polar Return to zero* (RZ), *Manchester*, dan *Differential Manchester*. *Polar Non Return to zero* menggunakan dua tingkat tegangan untuk merepresentasikan bilangan biner. Secara umum nilai tegangan positif merepresentasikan 1 dan nilai tegangan negatif merepresentasikan 0. Tidak ada nilai tanpa tegangan (0). Ada dua jenis *encoding* NRZ, yaitu NRZ-L (*Level*) dan NRZ-I (*Inverted*). Gambar 4.12 menggambarkan pengkodean menggunakan NRZ-L dan NRZ-I.

Gambar 4.12 Pengkodean dengan Non Return to Zero (NRZ) (a) Level (NRZ-L), (b) Inverted (NRZ-I)

Pengkodean *Non Return to Zero Level* (NRZ-L) adalah pengkodean menggunakan nilai tegangan positif yang merepresentasikan 1 dan nilai tegangan negatif yang merepresentasikan 0. Pengkodean ini digunakan pada standar RS-232 pada konektor DB-25. Gambar 4.13 menunjukkan gambar konektor DB-25 yang biasa digunakan pada printer *dot matrix*.

Gambar 4.13 Konektor DB-25

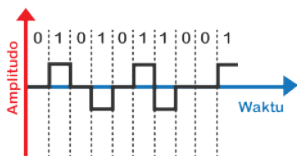
Sedangkan *Non Return to Zero Inverted* (NRZ-I) adalah pengkodean dengan melihat transisi dari tegangan. Jika sinyal berubah dari tegangan negatif ke tegangan positif atau sebaliknya maka representasi datanya adalah 1, sedangkan jika terjadi perubahan maka representasi datanya adalah 0. NRZ-I juga dapat menggunakan transisi sebaliknya, yaitu jika sinyal berubah dari tegangan negatif ke tegangan positif atau sebaliknya maka representasi datanya adalah 0, sedangkan jika terjadi perubahan maka representasi datanya adalah 1. Hal ini digunakan pada standar protokol universal serial bus (USB) yang banyak kalian gunakan pada laptop, ponsel, atau printer kalian. Gambar 4.14 menunjukkan gambar port dan konektor USB yang biasanya digunakan pada ponsel kalian.

(a) (b)

Gambar 4.14 (a) USB port, (b) Konektor USB Type C

Pengkodean Return to Zero (RZ) adalah pengkodean yang memperbaiki pengkodean NRZ, dimana NRZ memiliki kelemahan saat menentukan akhir dari runtunan bit atau

awal dari runtunan bit baru jika pengirim dan penerima tidak sinkron. RZ menggunakan tiga level tegangan, tegangan positif untuk merepresentasikan 1, tegangan negatif untuk merepresentasikan 0, dan tegangan 0 untuk transisi antar dua sinyal. RZ banyak digunakan sebagai protokol untuk komunikasi berbasis optik / cahaya, seperti infrared. Gambar 4.15 menunjukkan pengkodean Return to Zero (RZ).

Gambar 4.15 Return to Zero Encoding

c) *Bipolar Encoding*

Bipolar Encoding adalah pengkodean menggunakan tiga tingkat tegangan, yaitu: positif, negatif, dan nol. Tegangan nol merepresentasikan bilangan biner 0, sedangkan bilangan biner 1 direpresentasikan dengan tegangan positif atau negatif. Gambar 4.16 menunjukkan pengkodean dengan metode bipolar encoding.

Gambar 4.16 Bipolar Encoding

Ayo Lakukan!

Aktivitas Individu

Aktivitas JKI-K11-07-U: Pengkodean Sinyal Digital

Kalian akan melakukan aktivitas untuk membuat pengkodean data dari data digital menjadi sinyal digital. Kalian akan menyimpan sebuah *file* dari USB *flash drive* ke dalam laptop. *Flash drive* menggunakan protokol USB dimana pengiriman data dikirim menggunakan *line encoding* NRZ-I, dengan transisi tegangan mempresentasikan 0 sedangkan jika tidak ada transisi tegangan merepresentasikan 1.

Apa yang harus kalian lakukan?

Gambarkanlah sinyal digital dari data berikut ini:

a. 1 0 1 0 1 1 1 0 1 0 0 0

b. Data bilangan desimal 1028

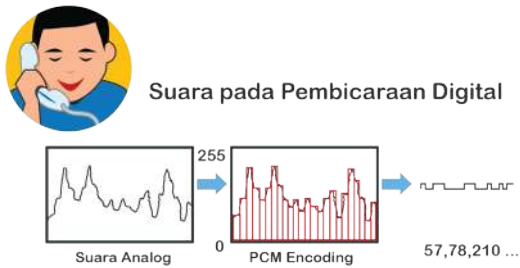
c. Data alfa numerik ‘a’ (kode ASCII 97)

Tahukah Kalian?

Block Coding menjamin akurasi data pada proses pengiriman data dengan memberikan tambahan bit yang disebut parity bit. *Parity bit* genap jika sinyal diberikan tambahan bit 1 atau 0 agar menjadi genap. Setelah menambahkan bit ini, sinyal siap untuk ditransmisikan.

2) Konversi Analog ke Digital

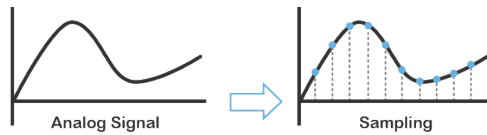
Saat kita berbicara dengan menggunakan *microphone* atau menggunakan kamera video, kita mengeluarkan suara yang merupakan sinyal analog. Suara manusia adalah sinyal kontinyu analog pada rentang frekuensi 0-4 KHz. Sedangkan sinyal digital berupa binary digit (bit) 1 dan 0. Untuk itulah diperlukan konversi suara manusia menjadi runtunan bit 1

dan 0 sebelum dikirimkan. Gambar 4.17 menggambarkan proses konversi suara manusia menjadi sinyal digital.

Gambar 4.17 Proses Konversi Suara menjadi Sinyal Digital

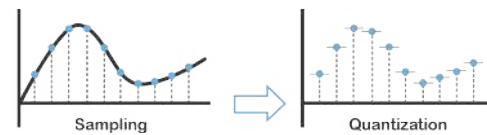
Untuk mentransmisikan data analog melalui media dalam bentuk sinyal digital, diperlukan teknik khusus yang disebut dengan modulasi. Salah satu teknik yang paling sering digunakan adalah pulse code modulation (PCM). PCM adalah teknik mengubah gelombang analog menjadi data digital dengan 3 langkah, yaitu: Sampling, Kuantisasi, dan Pengkodean.

a) *Sampling*

Sinyal analog di sampling pada interval waktu tertentu. Interval waktu ditentukan untuk mendapatkan *sampling rate* terbaik. *Sampling rate* terbaik menurut teorema Nyquist adalah paling tidak dua kali frekuensi tertinggi sinyal. Sebagai contoh range pendengaran manusia ialah 20Hz–20.000Hz, maka menurut Nyquist untuk menangkap frekuensi dengan lebar tersebut diperlukan 40.000Hz (40 kHz) sebagai *sampling rate*.

Sebagai contoh, suara pada telepon memiliki sampling rate 8 kHz, yang artinya suara di-*sampling* atau dicuplik sebanyak 8000 kali dalam satu detik. Sedangkan lagu yang disimpan dalam *compact disc audio* (CDA) memiliki *sampling rate* 44.1 KHz, yang berarti lagu ini dicuplik sebanyak 44.100 kali dalam satu detik yang menghasilkan suara yang hampir sama persis dengan aslinya.

Gambar 4.18 Sampling Sinyal Analog menjadi Digital

b) Kuantisasi

Setelah sampling dilakukan langkah selanjutnya adalah kuantisasi. Kuantisasi adalah proses untuk memberikan besaran satuan angka dari amplitudo data diskret hasil sampling.

Gambar 4.19 Kuantisasi Sinyal dari Sampling Sinyal

c) Pengkodean

Hasil kuantisasi selanjutnya dikodekan menjadi format digit biner.

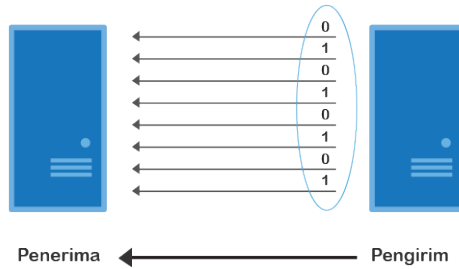
Gambar 4.20 Pengkodean Hasil Kuantisasi

3) Moda Transmisi

Sinyal dalam bentuk digital siap ditransmisikan. Data digital dapat ditransmisikan dengan dua mode yaitu paralel dan serial.

a) Transmisi Paralel

adalah transmisi yang dilakukan dengan membentuk sinyal digital dalam panjang yang tetap (*fixed*). Pengirim dan penerima terkoneksi secara paralel dengan jumlah jalur data yang sama. Pengirim akan mengirimkan bit data pada seluruh jalur (grup bit data/*data frame*) pada satu waktu tertentu. Transmisi paralel memiliki keunggulan kecepatan pengiriman, namun memiliki

kelemahan dengan banyaknya jalur yang harus dibuat. Ilustrasi transmisi paralel tampak pada gambar 4.21.

Gambar 4.21 Transmisi Paralel

b) Transmisi Serial

adalah transmisi sinyal digital melalui jalur tunggal secara berurutan dalam bentuk antrian. Transmisi ini hanya membutuhkan satu jalur data yang tampak pada gambar 4.22 berikut.

Gambar 4.22 Transmisi Serial

Transmisi serial dapat dilakukan dengan cara sinkron atau asinkron, Transmisi digital serial asinkron menggunakan cara pengiriman tanpa memperhatikan timing namun menggunakan pola (*pattern*) pada bit data. Sebagai contoh, prefix 0 bisa ditambahkan sebagai awal dari byte data yang dikirimkan, atau bit 1 ditambahkan sebagai penutup. *Data frame* biasanya memiliki celah/*gap* untuk membedakan antar byte data.

Transmisi digital serial sinkron menggunakan cara pengiriman yang serentak. Pengiriman ini mengandalkan timing yang tepat, dan tidak mengenal mekanisme awal atau akhir byte data. Bit data terkirim dalam burst mode dan tidak ada celah antar byte. Satu burst data bisa berisi beberapa

byte data. Keunggulan dari transmisi serial sinkron ini adalah kecepatannya yang tinggi dan tidak memiliki *overhead* karena penanda awal (*header*) dan penanda akhir (*footer*) dari bit data seperti pada transmisi asinkron.

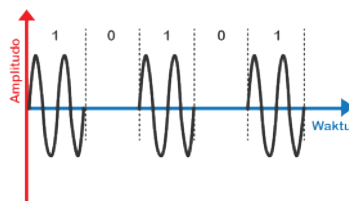
b. Transmisi Analog

Transmisi analog adalah proses pengiriman data secara analog. Data yang akan dikirimkan dapat berupa data analog atau digital. Data analog tidak dapat langsung dikirimkan, data ini harus diubah menjadi sinyal analog agar dapat ditransmisikan. Sedangkan untuk data digital juga harus diubah menjadi sinyal analog agar dapat dikirimkan.

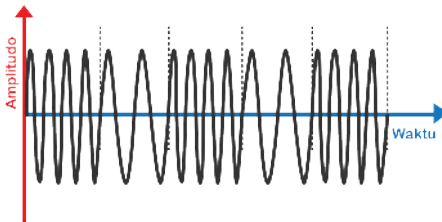
1) Konversi Data Digital ke Sinyal Analog

Agar dapat dikirimkan secara analog data digital harus dikonversikan menjadi sinyal analog. Sinyal analog adalah sinyal yang memiliki amplitudo, frekuensi, dan *fase*, sehingga konversi data digital ke analog dapat dilakukan dengan tiga cara ini, yaitu: *Amplitude Shift Keying*, *Frequency Shift Keying*, dan *Phase Shift Keying*.

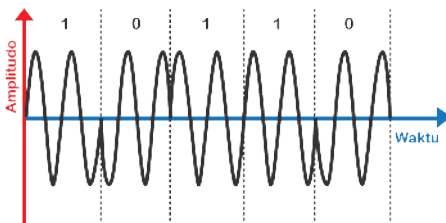
a) *Amplitude Shift Keying*

Amplitude Shift Keying menggunakan mekanisme mengubah digit biner menjadi amplitudo pada sinyal pembawa analog (*analog carrier signal*). Digit biner 1 akan direpresentasikan dengan gelombang dengan amplitudo tertentu, sedangkan digit biner 0 direpresentasikan tanpa amplitudo. Pada teknik ini frekuensi dan fase dari gelombang analog tetap sama. Gambar 4.23 menunjukkan amplitude shift keying untuk data biner 1 0 1 0 1.

Gambar 4.23 *Amplitude Shift Keying*

b) *Frequency Shift Keying*

Frequency Shift Keying adalah penciptaan sinyal analog menggunakan mekanisme perubahan frekuensi pada sinyal pembawa analog (*analog carrier signal*). Digit biner 1 akan direpresentasikan dengan gelombang dengan frekuensi tertentu yang biasanya lebih rapat (f1), sedangkan digit biner 0 direpresentasikan dengan gelombang dengan frekuensi yang lebih rendah (f2). Pada teknik ini amplitudo dan fase dari gelombang analog tetap sama. Gambar 4.24 menunjukkan *frequency shift keying* untuk data biner 1 0 1 0 1.

Gambar 4.24 *Frequency Shift Keying*

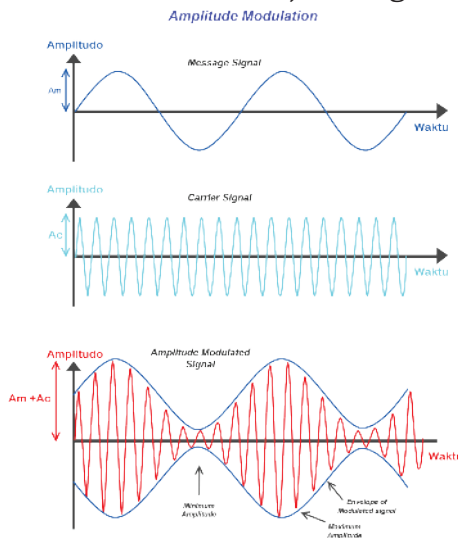
c) *Phase Shift Keying*

Phase Shift Keying adalah penciptaan sinyal analog menggunakan mekanisme perubahan fase pada sinyal pembawa analog (*analog carrier signal*). Digit biner akan direpresentasikan dengan gelombang dengan fase yang terbalik satu sama lain. Saat digit biner berubah maka fase dari gelombang akan berbalik. Pada teknik ini amplitudo dan frekuensi dari gelombang analog tetap sama. Gambar 4.25 menunjukkan phase shift keying untuk data biner 1 0 1 0 1.

Gambar 4.25 *Phase Shift Keying*

2) Konversi Data Analog ke Sinyal Analog

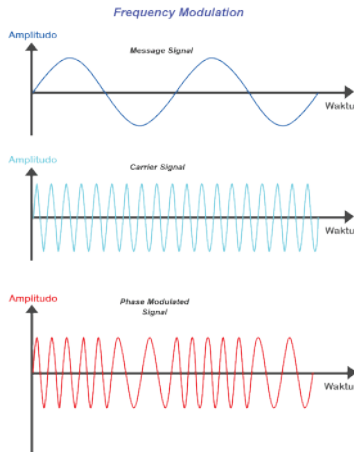
Agar dapat dikirimkan secara analog data analog harus dimodifikasi menjadi sinyal analog. Modifikasi ini biasa disebut dengan modulasi analog. Konversi data analog menjadi sinyal analog dapat dilakukan dengan tiga cara, yaitu: *Amplitude Modulation*, *Frequency Modulation*, dan *Phase Modulation*.

a) *Amplitude Modulation*

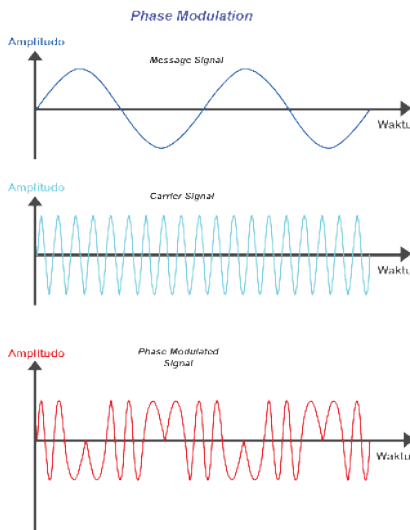
Pada modulasi ini, amplitudo dari sinyal pembawa analog akan dimodifikasi sesuai dengan perubahan data analog. Implementasi dari amplitude modulation dilakukan menggunakan pengali (*multiplier*) amplitudo. Amplitudo dari analog data akan dikalikan dengan gelombang pembawa sehingga menghasilkan modulasi amplitudo yang merefleksikan data analog yang ditransmisikan. Teknik modulasi ini adalah teknik awal yang ditemukan. Frekuensi dan fase dari gelombang pembawa tidak berubah. Gambar 4.26 menunjukkan gambar modulasi amplitudo.

Gambar 4.26 *Amplitude Modulation*

b) *Frequency Modulation*

Pada modulasi ini, *frequency* dari sinyal pembawa analog akan dimodifikasi sesuai dengan perubahan data analog. Amplitudo dan *Fase* dari gelombang modulasi tidak berubah. Gambar 4.27 menunjukkan modulasi frekuensi. *Frequency modulation*

digunakan pada *broadcasting* radio FM, radar, rekaman kaset magnetis, telemetri, dll. Keunggulan dari Frequency Modulation dibanding dengan modulasi lain adalah kualitas dari sinyal yang dihasilkan lebih baik dan memiliki derau *noise* yang lebih rendah.

Gambar 4.27 *Frequency Modulation*

c) *Phase Modulation*

Pada modulasi ini, fase dari sinyal pembawa analog akan dimodifikasi sesuai dengan perubahan gelombang data analog. Amplitudo dan frekuensi dari gelombang modulasi tidak berubah. Gambar 4.28 menunjukkan modulasi fase.

Gambar 4.28 *Phase Modulation*

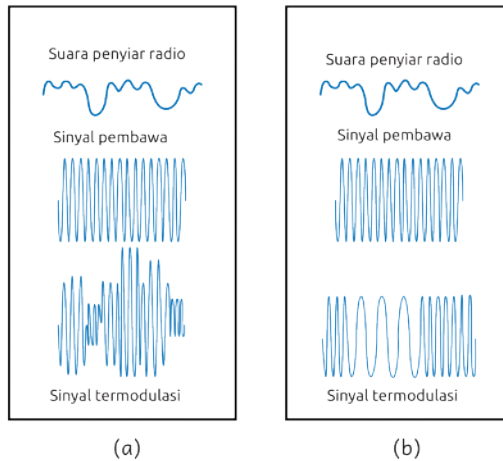
Ayo Lakukan!

Aktivitas Individu

Aktivitas JKI-K11-08-U: Menentukan Jenis Modulasi

Suara penyiar diterima oleh radio dalam bentuk sinyal yang termodulasi seperti gambar berikut:

(a)

(b)

Pertanyaan:

Menggunakan teknik modulasi apakah pengiriman suara penyiar radio tersebut?

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Setelah mengikuti pembelajaran transmisi data ini, apakah kalian menganggap transmisi digital atau analog yang telah kalian manfaatkan ketika mendengarkan lagu, dan menonton video di internet adalah sesuatu yang sulit?
2. Kesulitan apa yang kalian temukan pada materi ini?

3. Dengan pemahaman kalian saat ini, apakah kalian dapat memahami mengapa ketika kalian sedang melakukan *video conferencing* atau sedang menelepon melalui aplikasi, suara atau gambarnya bisa terputus-putus?
4. Apa ide kalian agar transmisi data bebas hambatan?
5. Dari aktivitas jaringan komputer dan internet di Kelas XI ini, aktivitas mana yang paling kalian sukai?

KEMENTERIAN PENDIDIKAN,
KEBUDAYAAN, RISET, DAN
TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas XI
Penulis : Irya Wisnubhadra
ISBN: 978-602-244-861-7

Pengembangan Aplikasi *Mobile* dengan *Library* Kecerdasan Artifisial

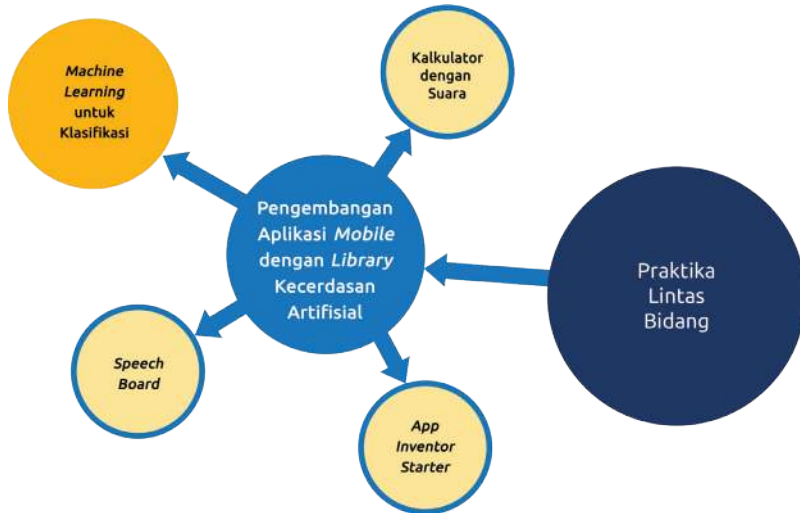
Tujuan Pembelajaran

Setelah mempelajari bab ini kalian diharapkan mampu mengembangkan sebuah aplikasi berbasis mobile yang merupakan implementasi kecerdasan buatan dengan langkah-langkah: analisis, identifikasi persoalan, perancangan, implementasi, pengujian, dan penyempurnaan. Selanjutnya kalian juga diharapkan mampu untuk mengomunikasikan produk aplikasi, dan manfaatnya secara lisan maupun tertulis.

Pertanyaan Pemantik

Hakikatnya kita sebagai manusia memiliki akal untuk berpikir dalam menyelesaikan masalah dan dapat terus meningkatkan kecerdasan

kita, lalu bagaimana dengan sebuah mesin buatan manusia? Dapatkah mesin tersebut berpikir dan terus meningkatkan kecerdasan berpikirnya seperti manusia?

Peta Konsep

Gambar 5.1 Peta Konsep Pengembangan Aplikasi *Mobile* dengan *Library* Kecerdasan Artifisial

Apersepsi

Saat ini mungkin sebagian dari kalian telah terbiasa menggunakan gawai (*mobile phone*) berupa ponsel, tablet, atau yang lain. Mungkin sebagian dari kalian juga telah terbiasa menggunakan aplikasi *mobile*. Aplikasi *mobile* adalah salah satu bentuk artefak komputasional yang bermanfaat bagi kehidupan masyarakat di era digital saat ini. Aplikasi ini sama dengan aplikasi lain yang telah kalian kembangkan pada jenjang sebelumnya namun dirancang untuk dapat berjalan pada ponsel (*mobile phone*). Pengembangan aplikasi *mobile* tidaklah sesulit yang dibayangkan. Pada bab ini kalian akan mempelajari bagaimana cara mengembangkan aplikasi *mobile*, dan dilanjutkan dengan penggunaan *library* atau komponen kecerdasan artifisial. *Library* adalah modul program dengan fungsi tertentu yang sudah dikemas sehingga siap dipakai

tanpa pemrogram pemakainya perlu mengimplementasi kodenya. Memakai *library* ini dapat diibaratkan kalian menggunakan ponsel atau komputer dengan mudah dan nyaman tanpa perlu tahu betapa rumit isi di dalamnya.

Kata Kunci

Aplikasi *mobile*, Kecerdasan Artifisial, *Machine Learning*, Klasifikasi.

A. Pengembangan Aplikasi *Mobile* dengan *App Inventor*

Saat ini kehidupan sehari-hari manusia banyak dibantu oleh aplikasi atau perangkat lunak yang terpasang pada ponsel pintar, komputer, atau diakses secara *online* lewat peramban. Aplikasi tersebut diantaranya adalah aplikasi perkantoran, aplikasi bertukar pesan, pemutar lagu, aplikasi desain, pengolah akuntansi, dll. Aplikasi dapat dibedakan berdasarkan platform pengembangan dan penggunaannya, yaitu aplikasi *desktop*, aplikasi *web*, dan aplikasi *mobile*. Penulisan aplikasi sering disingkat menjadi apps.

1. *Desktop Apps*

Aplikasi *desktop* adalah aplikasi yang dikembangkan dengan tujuan implementasi pada komputer *desktop* atau piranti lokal komputer. Aplikasi ini harus dipasang pada piranti lokal komputer, dan setelah terpasang aplikasi ini akan berada pada memori dari piranti lokal.

2. *Web Apps*

Aplikasi berbasis web adalah aplikasi yang dikembangkan dengan tujuan dapat diakses menggunakan koneksi jaringan komputer dan internet menggunakan protokol http. Aplikasi ini tidak terpasang pada piranti atau komputer *desktop* lokal, namun terpasang pada *server* tertentu. Aplikasi ini kebanyakan diakses menggunakan *browser*, namun ada juga

yang berbentuk *client side* dimana ada program kecil yang terpasang pada piranti lokal, tetapi proses komputasi utama dilakukan di *server*.

3. *Mobile Apps*

Aplikasi mobile juga disebut dengan mobile apps adalah aplikasi yang dirancang untuk dapat dieksekusi pada piranti *mobile* seperti ponsel, tablet, atau *smart watch*. *Mobile* memiliki arti mudah bergerak. Aplikasi *mobile* dapat dipasang pada ponsel, tablet, atau gadget lainnya. Aplikasi ini berkembang pesat karena kemudahan penggunaan piranti *mobile*, dan dapat diintegrasikan dengan sistem lain yang ada pada piranti mobile seperti GPS, kamera, sidik jari, dll. Saat ini tersedia jutaan aplikasi *mobile* yang ada di platform pendistribusian aplikasi digital yaitu *Play Store* ataupun *Apps Store*.

Pengembangan aplikasi banyak dibantu oleh perkakas pengembangan yang disebut dengan *Integrated Development Environment* (IDE). IDE membantu kemudahan dan efektifitas pengembangan perangkat lunak. Salah satu IDE yang digunakan untuk pengembangan aplikasi mobile adalah App Inventor.

4. *App Inventor*

App Inventor adalah perangkat lunak IDE terintegrasi yang berbasis *web*. App Inventor pada awalnya dikembangkan oleh Google, yang saat ini dipelihara oleh Massachusetts Institute of Technology (MIT). App Inventor memungkinkan pemrogram komputer pemula dapat membuat *mobile apps* diatas OS Android maupun iOS. Aplikasi App Inventor bersifat *open source* dan *free*.

App Inventor memiliki antarmuka berbasis grafis dan memiliki tampilan yang mirip dengan bahasa pemrograman block Scratch/Blockly yang telah kalian pelajari di kelas 7 dan 8. Dengan App Inventor kalian dapat membuat program dengan cara seret lepas (*drag and drop*) komponen-komponennya. App

Inventor sebagai perkakas, terus dikembangkan kecanggihannya melalui riset intensif di bidang *educational computing*. App Inventor mendukung penggunaan *cloud data* dengan *Firebase* dan *Firebase Realtime Database*.

App Inventor dapat diakses melalui peramban dari situs https://ai2.appinventor.mit.edu. Tampilan awal App Inventor tampak pada gambar 5.2 berikut:

Gambar 5.2 Tampilan Awal App Inventor

App Inventor memiliki banyak komponen yang dapat digunakan dalam pembuatan aplikasi. Komponen tersebut dikelompokkan dalam *User Interface Components*, *Layout Components*, *Media Components*, *Drawing and Animation Components*, dll. *User Interface Components* memiliki komponen-komponen yang berhubungan dengan antarmuka pengguna, seperti: *Button* (Tombol), *CheckBox*, *DatePicker*, *Image*, dll. Masing-masing komponen memiliki *methods*, *events*, dan *properties* yang digunakan untuk memanipulasi komponen tersebut.

Properties pada komponen adalah atribut yang mendeskripsikan sifat dari komponen, misalnya lebar tombol, warna dari teks, dll. *Properties* biasanya dapat dibaca dan di set, namun ada juga *properties* yang hanya bisa dibaca.

Methods adalah fungsi yang dapat dikenakan pada komponen yang memilikinya, *methods* dapat digunakan untuk mengatur *properties*.

Events adalah kejadian yang terjadi karena pemanggilan *methods*, seperti aksi *mouse click* yang menghasilkan *mouse event* yang menyebabkan suatu fungsi/prosedur akan dieksekusi.

Sebagai contoh, komponen *Button* (Tombol) yang memiliki kemampuan untuk mendeteksi penekanan tombol, memiliki *properties*: warna *background* tombol yang dapat diubah sesuai keinginan, tombol dapat di set *enabled* (aktif) atau tidak aktif, dan *font* yang dapat diset *italic*, *bold*, dll. *Properties* tersebut dapat dimanipulasi pada *Designer Editor* atau pada Blocks Editor. Dalam Tabel 5.1. berikut memberikan berbagai contoh komponen, methods, event, dan properties dari komponen pada App Inventor.

Tabel 5.1. Komponen, Properties, Events, dan Methods pada App Inventor

Komponen	Properties	Events	Methods
User Interface Components			
Button (Tombol): yang memiliki kemampuan mendeteksi penekanan (klik) dari user	BackgroundColor(Color)	Click()	(tidak ada)
	Enabled(Boolean)	GotFocus()	
	FontSize(Number)	LongClick()	
	FontBold(Boolean)	LostFocus()	
	FontItalic(Boolean)	TouchDown()	
	Image(Text)		
User Interface Components			
TextBox: Kotak tempat user mengisi teks	Text(Text)	GotFocus()	HideKeyboard()
	BackgroundColor(Color)	LostFocus()	RequestFocus()
	Enabled(Boolean)		
	FontSize(Number)		
	FontBold(Boolean)		
	FontItalic(Boolean)		
	Multiline(Boolean)		

Komponen	Properties	Events	Methods
Label: komponen untuk menampilkan teks	Text(Text)	(tidak ada)	(tidak ada)
	BackgroundColor(Color)		
	Enabled(Boolean)		
	FontSize(Number)		
	FontBold(Boolean)		
	FontItalic(Boolean)		
	TextColor(Color)		
Layout Components			
Horizontal Arrangement: Komponen ini memformat elemen yang diletakkan pada layar akan tertampil secara horizontal dari kiri ke kanan	AlignHorizontal(Number)	(tidak ada)	(tidak ada)
	AlignVertical (Number)		
	BackgroundColor(Color)		
	Image(Text))		
Media Components			
TextToSpeech: Komponen yang akan mengubah teks menjadi suara	Language(Text)		
	Country(Text)		
	SpeechRate(Number)		

Ayo Berlatih!

Aktivitas Individu

Aktivitas PLB-AI-K11-01: Starter App Inventor – Halo Dunia, dengan Text to Speech

Melalui aktivitas ini kalian akan mulai belajar cara pembuatan aplikasi mobile dengan App Inventor yang dapat dijalankan di ponsel Android. Aplikasi ini mampu mengubah teks yang dituliskan pada aplikasi ponsel menjadi suara yang terdengar melalui pengeras suara di ponsel.

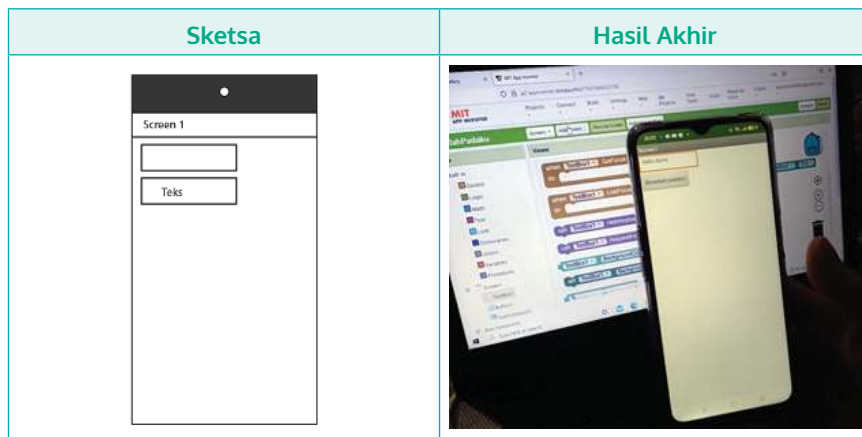
Persiapan:

Pada aktivitas ini kalian memerlukan komputer yang terkoneksi dengan internet dan terpasang perangkat lunak MIT AI2 Companion, ponsel atau tablet dengan sistem operasi Android/iOS. Namun jika ponsel tidak tersedia maka kalian dapat menggunakan emulator ponsel yang akan muncul pada layar komputer kalian. Kalian harus melakukan pengaturan khusus untuk emulator ini.

Prasyarat:

Kalian sebagai peserta didik harus telah memahami pemrograman dengan Scratch/Blockly yang dipelajari di SMP.

Deskripsi Produk :

Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:

Spesifikasi Aplikasi:

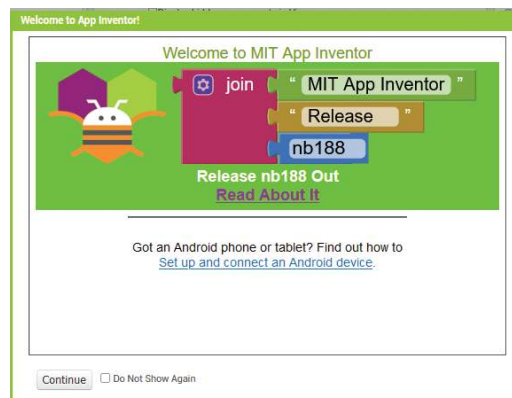
- **Input**: Pengguna mengetikkan "Halo Dunia" pada *textbox* dan mengetuk tombol di bawah textbox
- **Proses**: Aplikasi mengubah teks yang ditulis pada program menjadi suara
- **Output**: Aplikasi akan memperdengarkan suara lewat speaker ponsel

Langkah-langkah:

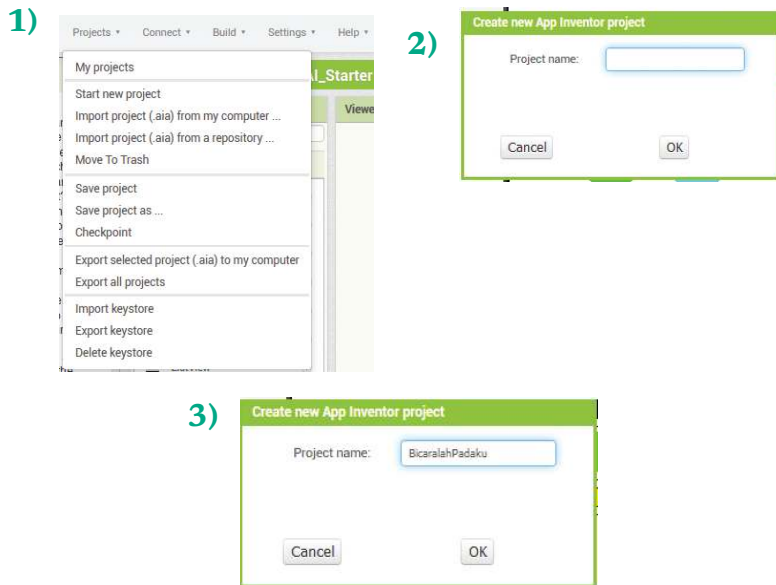
1. Persiapan:

a. Masuk/*login* ke situs App Inventor (https://appinventor.mit.edu/) dengan menggunakan akun google kalian.

*jika belum memiliki akun google, kalian dapat menggunakan milik orang tua, kakak atau minta bantuan gurumu untuk mendaftarkannya.

b. Klik *Continue* saat layar pembuka (*splash screen*) muncul.

2. Pengkodean:

a. Buat proyek baru dengan memilih menu **Projects** lalu klik **Start new project**. Lalu akan muncul *menu pop-up* untuk mengisi nama proyek, kalian beri nama proyek baru tersebut dengan "BicaralahPadaku" (tanpa spasi). Perlu diketahui setiap kalian membuat aplikasi di App Inventor, aplikasi tersebut disimpan dalam sebuah proyek yang berisi semua

file terkompilasi ke dalam sebuah *executable file*. *File* tersebut dapat berisi kode sumber, ikon, gambar, suara, *file data*, dsb.

b. Perancangan *User Interface* (UI):

Saat mengembangkan aplikasi mobile kalian, pengembangan dilakukan dengan merancang *User Interface* (UI) dan perancangan blok kode.

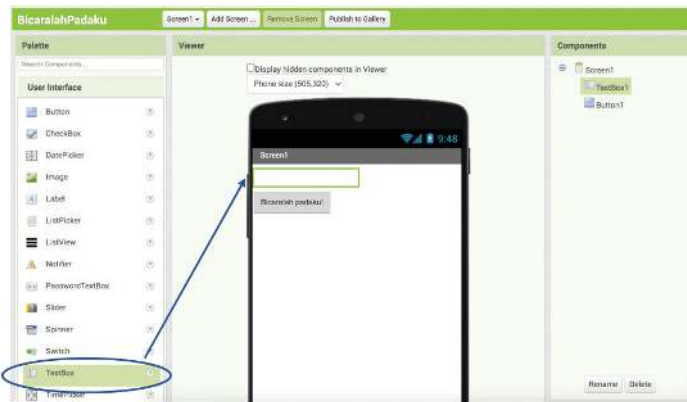
Perancangan UI dilakukan dengan menggunakan tampilan Designer, yang tampil dengan mengklik tombol *designer* pada bagian kanan atas. Tampilan Designer App Inventor memiliki empat kolom. Kolom *Palette* merupakan tempat komponen-komponen yang tersedia dari App Inventor, kolom *Viewer* merupakan kolom untuk perancangan UI aplikasi, kolom *Components* berisi komponen-komponen yang digunakan pada proyek, dan kolom *Properties* yang merupakan kolom untuk melakukan pengaturan terhadap komponen-komponen yang digunakan.

Perancangan blok kode dilakukan dari Tampilan Blocks yang tampil dengan mengklik tombol Blocks pada

bagian kanan atas disamping tombol Designer yang akan dijelaskan pada kemudian.

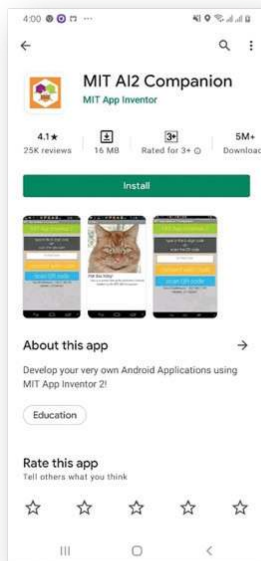
Langkah perancangan UI:

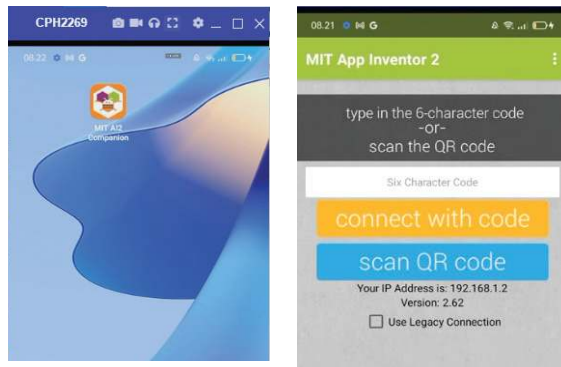
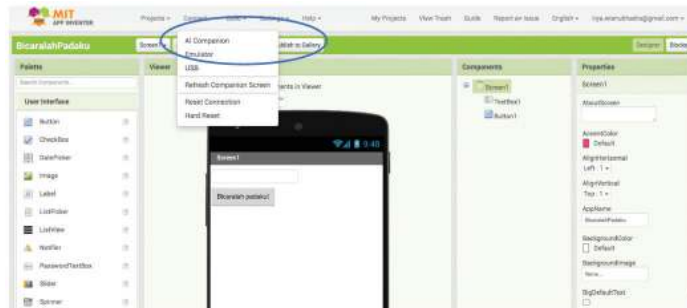
1) Tambahkan tombol Button pada Viewer, dengan seret dan lepaskan (*drag and drop*) dari kolom Palette ke kolom Viewer. Sebuah Button dengan nama Button1 akan tercipta. Nama Button1 dapat diganti dengan nama lain yang sesuai, namun pada latihan ini kalian masih menggunakan nama default.

2) Tambahkan TextBox dengan *drag and drop* ke area Viewer. Secara otomatis TextBox dengan nama default TextBox1 akan muncul.

3. Persiapan Pengujian

Aplikasi yang dikembangkan adalah aplikasi mobile yang berjalan pada piranti *mobile phone*, sehingga pengujian idealnya dilakukan dengan menguji dengan ponsel secara *live*, langkah persiapan pengujian dilakukan dengan langkah:

a. Sambungkan App Inventor pada komputer dengan *smartphone* kalian dengan menyambungkannya melalui kabel USB atau menggunakan perangkat wifi.

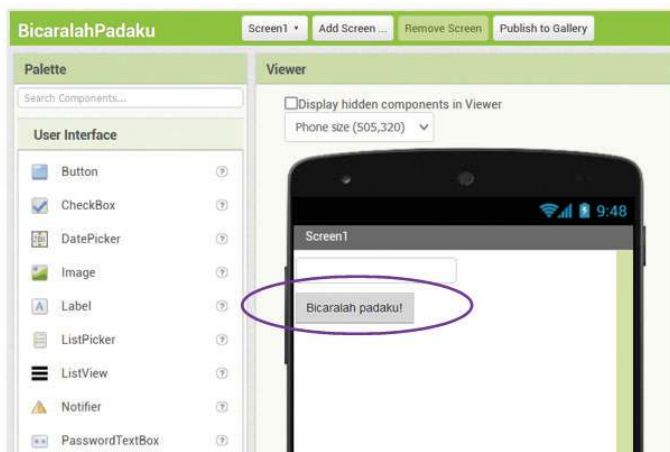
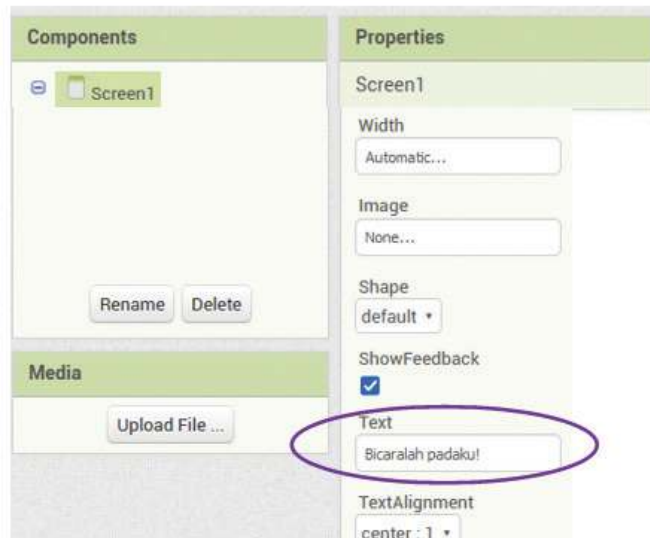
b. Unduh dan pasang/*install* MIT AI2 Companion di PlayStore/App Store pada ponsel kalian.

c. Pengunduhan dan pemasangan AI Companion memerlukan pengaturan (*setting*) pada ponsel. Lakukan centang (*check*) untuk membolehkan "*Unknown Sources*" pada menu "*Security*" pada ponsel. Selanjutnya, Scan QR code untuk mengunduh langsung MIT AI2 Companion atau klik link "Need help finding the Companion App?". Setelah selesai diunduh, pasang/*install* aplikasi MIT AI2 Companion.

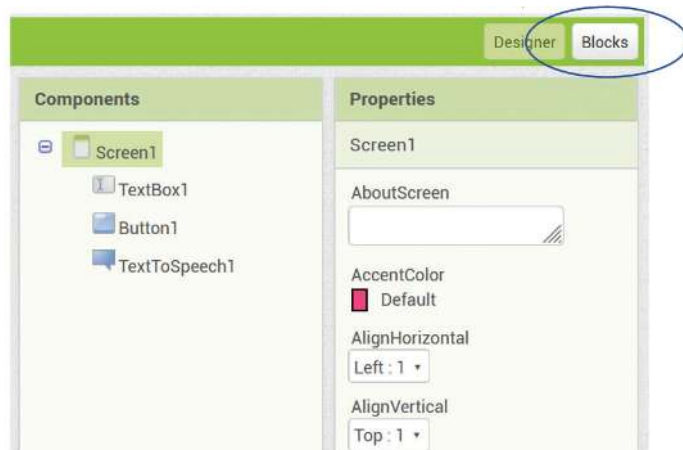
   Berikut tampilan App Inventor, jika sukses pemasangannya pada ponsel:

   Berikut ini tampilan App Inventor pada komputer/laptop. Untuk menghubungkan aplikasi App Inventor pada ponsel dengan App Inventor yang digunakan pada komputer/laptop. Caranya dengan pilih menu Connect pada App Inventor komputer/laptop kalian, lalu klik menu AI Companion.

   Berikut ini tampilan menu *pop-up* AI Companion pada komputer/laptop, akan muncul dalam *pop-up* kode 6 digit dan QR Code yang dapat digunakan untuk menghubungkan

kedua perangkat. Kalian dapat mengisikan kode 6 digit atau QR code yang dapat dipindai menggunakan ponsel untuk menghubungkan kedua perangkat tersebut.

Selanjutnya akan muncul Progress Bar dalam komputer/laptop kalian sebagai tanda proses menghubungkan kedua perangkat. Setelah itu, jika instalasi sukses maka kalian dapat melihat app kalian di ponsel. Jika kalian menambahkan komponen lain pada app kalian di komputer, maka perubahan akan terjadi juga di ponsel. Selamat, ponsel kalian siap digunakan untuk pengujian aplikasi.

4. Lanjutan Pengkodean:

a. Masih pada Designer view, ubahlah teks pada TextBox1, menjadi "Bicaralah padaku!" pada kolom properties.

b. Tambahkan komponen TextToSpeech pada Viewer, dengan cara drag and drop komponen tersebut ke kotak Viewer. Pilih komponen dari menu Pallete > Media > TextToSpeech.

c. Pengkodean Blok: Setelah komponen TextToSpeech ditambahkan, selanjutnya kalian lanjutkan pengkodean blok dengan masuk ke mode editor Blocks. Beralih ke mode editor Blocks dilakukan dengan menekan tombol Blocks pada pojok kanan halaman.

Editor Blocks adalah tempat untuk menyusun program dari aplikasi. Pada editor ini terdapat blok Built-in yang telah tersedia dan dapat digunakan untuk menangani operasi Control, Logic, Math, dll. Blocks merupakan ruang kerja yang digunakan untuk menyusun program aplikasi.

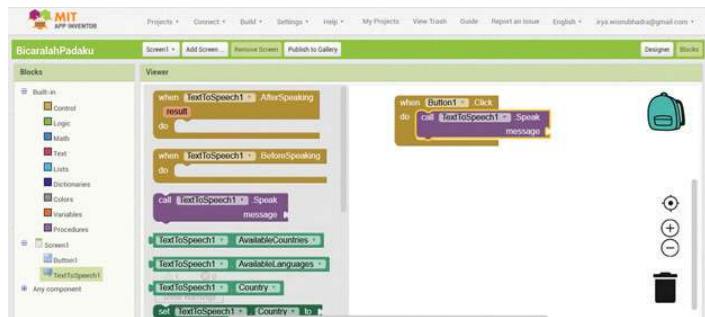
Pengkodean Blok selanjutnya dilakukan dengan langkah berikut ini:

1) **Buatlah *event* ketika tombol Button1 ditekan, dengan cara:**
   a) Klik tombol Button1 pada kolom Blocks
   b) Pilih blok when Button1.Click pada Viewer
   c) *Drag and drop* pada Viewer yang kosong, yang hasilnya tampak seperti pada gambar berikut:

2) **Tambahkan blok call TextToSpeech1.Speak dari komponen TextToSpeech1 ke blok when Button1. Click dengan cara:**

a) Klik TextToSpeech1 pada kolom Blocks

b) Pilih blok call TextToSpeech1.Speak di kolom Viewer

c) *Drag and drop* blok call TextToSpeech1.Speak, pada kolom Viewer yang kosong, yang hasilnya tampak seperti pada gambar berikut:

3) **Tambahkan blok TextBox1.Text dari komponen TextBox1 ke blok call TextToSpeech1.Speak dengan cara:**

a) Klik TextBox1 pada kolom Blocks

b) Pilih blok call TextBox1.Text di kolom Viewer

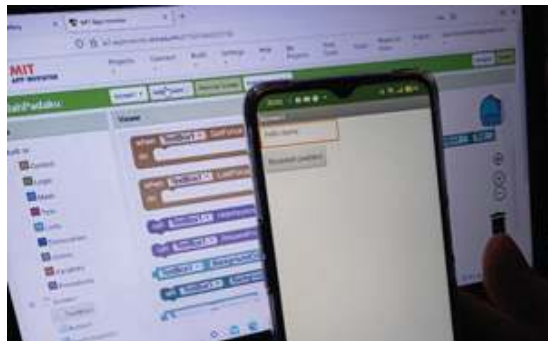
c) *Drag and drop* blok TextBox1.Text ke blok call TextToSpeech1.Speak pada kolom Viewer, yang hasilnya tampak seperti pada gambar berikut:

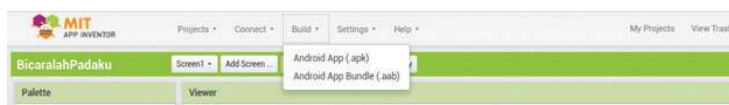
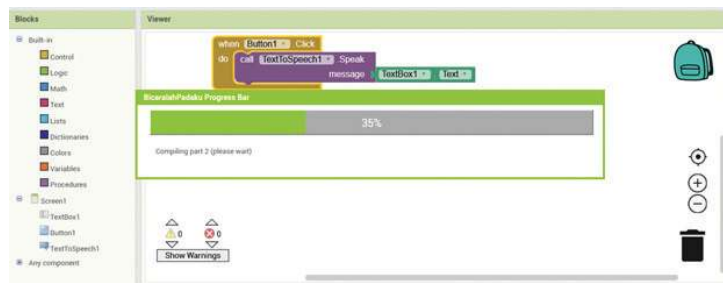
4) **Simpan file proyek kalian dengan memilih menu Projects > Save Project**

5. Pengujian Aplikasi:

Setelah selesai dengan langkah 4, langkah berikutnya adalah pengujian aplikasi. Pengujian aplikasi dilakukan dengan menuliskan teks "Halo Dunia" pada TextBox1, dan mengetuk tombol "Bicaralah padaku!" pada ponsel yang telah tersambung dengan komputer/laptop sebelumnya pada langkah 3. Jika speaker pada ponsel mengeluarkan suara Halo Dunia maka kalian telah berhasil membuat aplikasi mobile pertama kalian. Kalian saat ini menggunakan TextToSpeech, yang merupakan library App Inventor yang berfungsi mengubah teks menjadi suara, meniru kalian (manusia) membaca teks dan mengucapkannya! Kalian tinggal memakai, dan tidak perlu tahu betapa rumitnya program di dalamnya.

Setelah itu cobalah mengetikkan teks yang berbeda, atau dengan bahasa yang berbeda, misalnya bahasa Inggris atau Perancis, dan tekan tombol Bicaralah Padaku. Apa yang terjadi?

Berikut ini foto dari aplikasi BicaralahPadaku yang telah selesai dibuat.

Aplikasi yang telah kalian kembangkan dapat diunduh dan kalian bagikan ke teman dan orang tua, dengan cara:

a. Pilih menu Build dan pilih Android App (.apk).

b. Klik Download .apk now, dan file .apk akan terunduh. File .apk adalah file paket android (*Android Application Package*) yang digunakan untuk mendistribusikan aplikasi, file dapat digandakan dan dipasang pada piranti mobile kalian dan teman-teman kalian.

Ayo Kembangkan!

Aktivitas Kelompok

Aktivitas PLB-AI-K11-02: Speechboard

Pada aktivitas ini kalian akan belajar untuk mengembangkan aplikasi yang mampu memainkan sebuah rekaman pidato dengan menyentuh sebuah gambar.

Kebutuhan Alat dan Bahan:

1. Komputer yang terkoneksi dengan internet, ponsel/tablet dengan sistem operasi Android atau iOS. Komputer juga harus terpasang perangkat lunak MIT AI2 Companion. Jika ponsel tidak tersedia dapat digunakan emulator pada komputer.

2. Gambar proklamator pada saat mengucapkan proklamasi kemerdekaan Indonesia (*file type* .jpg) dan rekaman pidato pembacaan teks proklamasi pada tanggal 17 Agustus 1945 (*file type* .mp3). File dapat diunduh di https://static.buku.kemdikbud.go.id/content/media/rar/Informatika_XI.rar

Prasyarat:

1. Kalian harus telah memahami materi pemrograman menggunakan Scratch/Blockly yang dipelajari pada jenjang SMP.
2. Kalian memahami cara mengunduh file gambar dan suara dari internet, dan menyimpannya di komputer.

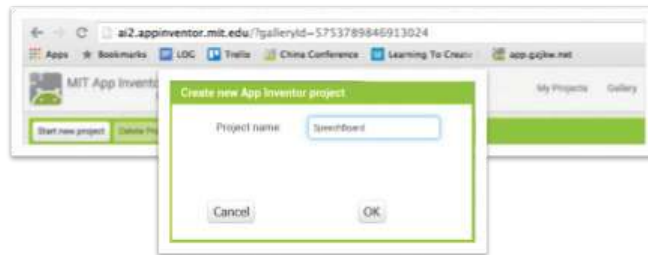
Deskripsi Produk :

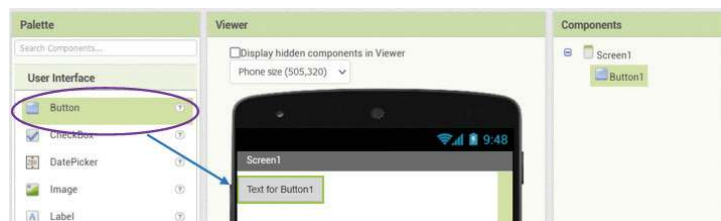
Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:

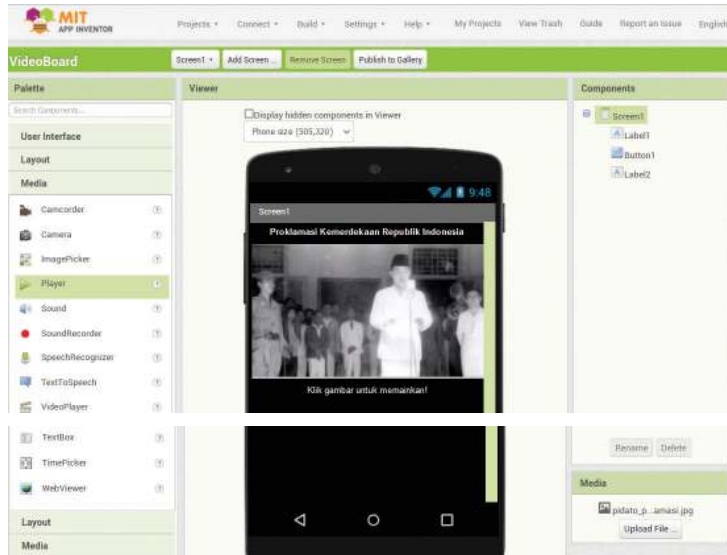
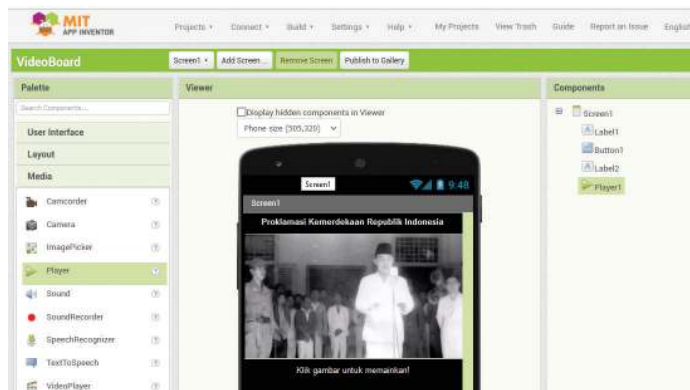
Spesifikasi Aplikasi:

- *Input*: Pengguna mengetuk gambar Sang Proklamator Indonesia di ponsel
- Proses: Aplikasi memainkan rekaman pidato proklamasi kemerdekaan Indonesia
- *Output*: Aplikasi memperdengarkan suara rekaman lewat speaker ponsel

Langkah langkah:

1. Persiapan:
   a. Buka aplikasi App Inventor dengan mengakses https://ai2.appinventor.mit.edu
   b. Mulailah membuat proyek baru, namailah proyek dengan nama "**SpeechBoard**"

2. Perancangan User Interface (UI):
   a. Berikut tampilan awal dari proyek,

   b. Tambahkan Button pada layar dengan cara seret dan lepaskan (*drag and drop*)

   c. Ubahlah *background* Button pada *screen* dengan gambar Proklamasi Kemerdekaan RI, dengan sebelumnya mengunggah gambarnya (*file* .png).

d. Tambahkan dua label dengan teks “Proklamasi Kemerdekaan Republik Indonesia” dan “Klik gambar untuk memainkan!” pada bagian atas dan bawah *button*. Seperti contoh berikut:

e. Tambahkan komponen Player dari kolom Palette > Media, sehingga Player1 tercipta, dan upload file pidato kemerdekaan (*file* .mp3) sebagai Source dari Player1.

3. Pengkodean Blok:
   a. Berikutnya, kalian harus menambahkan blok kode dengan beralih ke mode Blocks.
   b. Tambahkan kode program untuk memainkan *file* pidato dengan blok when Button1.Click.
   c. Isi blok call Player1.Start ke dalam blok when Button1. Click.

   d. Dan terakhir jangan lupa untuk menyimpan proyek (save project) kalian.
4. Pengujian:

   Ujilah kode program dengan mengetuk tombol pada ponsel untuk memainkan rekaman pidato, seperti pada langkah pengujian aktivitas PLB-AI-K11-01. Jika program telah berhasil memperdengarkan suara proklamasi, maka program kalian telah sesuai dengan spesifikasi, dan lanjutkan dengan aktivitas pengembangan.

Ayo Kembangkan!

Aktivitas Kelompok

Deskripsi Proyek:

Kalian diharapkan mengembangkan proyek perangkat lunak berbasis mobile yang dinamakan speechboard, dengan melakukan modifikasi aplikasi tersebut dengan menambahkan beberapa pidato dari para pahlawan Indonesia, misalnya Ki Hajar Dewantara, Sutomo, dll. Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:

Spesifikasi Aplikasi:

- **Input**: Pengguna mengetuk salah satu gambar pahlawan Indonesia di ponsel
- **Proses**: Aplikasi memainkan rekaman pidato yang terkenal dari para pahlawan tersebut
- **Output**: Aplikasi akan memperdengarkan rekaman pidato dari pahlawan yang dipilih lewat speaker ponsel

Gunakan Lembar Kegiatan Peserta Didik saat kalian mengembangkan aplikasi untuk berbagi peran dan tugas, berikut:

LKPD-01 Format Lembar Kegiatan Peserta Didik Pengembangan Aplikasi

Peran	Penanggung Jawab
Analis Program: a. Deskripsi Produk b. Spesifikasi Aplikasi c. Kebutuhan resource: file, alat, dll	
Perancang User Interface (UI)	
Pemrogram Kode	
Penguji Program	
Pemapar Presentasi	

Spesifikasi (Deskripsi Produk, Fungsionalitas Aplikasi, Kebutuhan *Resource*)

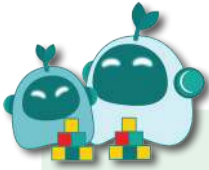
Rancangan *User Interface* (UI)

Kode Program

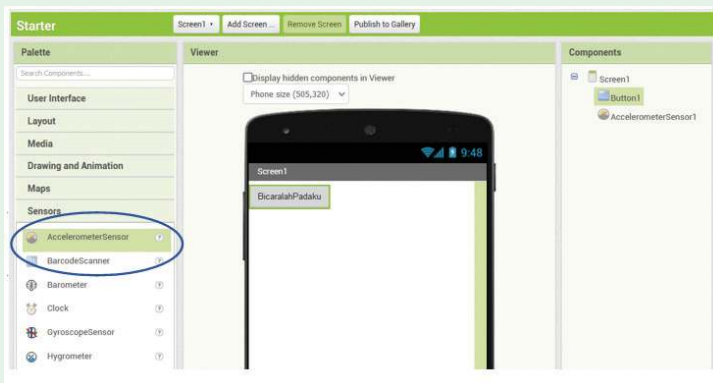
Pengujian

<table>
<tr><th colspan="3">Diisi saat perencanaan</th><th colspan="3">Diisi setelah Pengujian</th></tr>
<tr><th rowspan="2">No.</th><th rowspan="2">Fitur</th><th rowspan="2">Dikerjakan Oleh</th><th colspan="2">Sesuai dengan spesifikasi</th><th rowspan="2">Keterangan Hasil Pengujian</th></tr>
<tr><th>YA</th><th>TIDAK</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Tabel 5.2 Format LKPD-01 Pengembangan Aplikasi Aktivitas PLB-AI-K11-02

Selanjutnya, kembangkanlah aplikasi untuk dapat menghentikan suara (pause) ketika gambar diketuk, dan memperdengarkan suara kembali ketika diketuk lagi (toggle). Setelah itu, kembangkan juga dengan hanya boleh satu player yang hidup pada suatu saat tertentu. Jangan lupa setelah ditambah fungsionalitasnya ujilah kembali aplikasi kalian tersebut.

Pengayaan

Untuk meningkatkan pemahaman dan kemampuan kalian dalam materi Pengembangan Aplikasi Mobile ini, kalian dapat mengikuti aktivitas pengayaan berikut. Aplikasi yang telah kalian buat pada aktivitas PLB-AI-K11-01 dan PLB-AI-K11-02 dapat dikembangkan menjadi aplikasi lain, misalnya:

Aplikasi pada aktivitas PLB-AI-K11-01 dapat dikembangkan dengan memperdengarkan suara teks saat ponsel digerakkan naik turun. Aplikasi ini akan mengakses sensor accelerometer, sehingga teks akan disuarakan ketika ponsel digerakkan naik turun.

Mengembangkan aplikasi pemanggil nama peserta didik, sehingga guru dapat memanggil peserta didik dengan menekan tombol pada aplikasi di ponsel.

Ayo Renungkan!

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Setelah mengikuti pembelajaran ini, bagaimana pendapat kalian tentang pengembangan aplikasi mobile? Mudah bukan? Apakah kemudian kalian ingin belajar lebih banyak tentang pengembangan aplikasi mobile ini?
2. Kesulitan apa yang kalian temukan pada materi ini?
3. Dengan pemahaman saat ini, ide pengembangan apa yang kalian sarankan untuk membuat aplikasi yang lebih menarik?
4. Setelah kalian berhasil membuat aplikasi, apakah kalian merasa lebih familiar dengan teknologi? Apakah kalian ingin mengirimkan aplikasi kalian yang kalian buat ke orang tua atau teman kalian?

B. Pengembangan Aplikasi Kecerdasan Artifisial dengan App Inventor

Tahukah kalian saat ini teknologi kecerdasan artifisial telah banyak diimplementasikan pada kehidupan kita sehari-hari. Kalian mungkin pernah mendengar Google Assistant, Apple Siri, Amazon Alexia yang merupakan aplikasi asisten pribadi yang dapat melakukan pekerjaan tertentu dengan perintah menggunakan suara. Saat ini banyak perusahaan di Indonesia yang menggunakan *chatbot* untuk berinteraksi dengan konsumen secara otomatis, atau ketika kalian menggunakan youtube maka akan muncul video rekomendasi yang sesuai dengan kesukaan kalian.

Nama-nama produk diatas adalah contoh-contoh produk hasil dari kecerdasan artifisial, dan masih banyak contoh lain yang digunakan di industri dalam bentuk robot otomasi industri, robot penjelajah ruang angkasa, dll.

Kecerdasan Artifisial

Kecerdasan artifisial atau *Artificial Intelligence* (AI) adalah kecerdasan yang dimiliki oleh sistem atau mesin atau komputer. AI mampu untuk melakukan tugas yang umumnya terkait dengan kemampuan makhluk cerdas. Istilah ini sering diterapkan pada proyek pengembangan sistem yang memiliki sifat intelektualitas manusia, seperti kemampuan untuk menalar, menemukan makna, melakukan generalisasi, atau belajar dari pengalaman masa lalu. Sejak perkembangan komputer digital pada tahun 1940-an, AI telah telah banyak diimplementasikan untuk melakukan tugas yang kompleks seperti, misalnya, menemukan bukti untuk teorema matematika atau bermain catur dengan sangat mahir.

Namun, meskipun kemajuan terus-menerus dalam kecepatan pemrosesan komputer dan kapasitas memori, belum ada program yang dapat menandingi fleksibilitas manusia dalam domain yang lebih luas atau dalam tugas-tugas yang membutuhkan banyak pengetahuan sehari-hari. Di sisi lain, beberapa program telah mencapai tingkat kinerja yang sangat impresif yang dapat menggantikan para ahli dan profesional manusia dalam melakukan tugas-tugas tertentu tertentu, seperti diagnosis medis, mesin pencari komputer, dan pengenalan suara atau tulisan tangan.

Kecerdasan Artifisial kemudian berkembang dengan memunculkan berbagai subbidang yaitu:

a. ***Machine Learning***: *Machine Learning* adalah mesin pembelajar yang mampu melakukan pembuatan model analitik secara otomatis. Mesin ini menggunakan beberapa metode berbasis statistik, jaringan saraf, fisika, dll untuk menemukan *insight* (wawasan) tersembunyi dari data tanpa diprogram secara eksplisit. Mesin ini mampu mengambil kesimpulan secara otomatis

b. ***Deep Learning***: *Deep Learning* adalah mesin pembelajar dengan pembelajaran mendalam menggunakan jaringan syaraf tiruan dengan ukuran dan lapisan unit pemrosesan yang besar. *Deep Learning* memanfaatkan kemajuan dalam kemampuan komputasi dari perangkat komputer yang semakin cepat dan *algoritma training* (pelatihan) yang terus meningkat kinerjanya untuk mengenali pola kompleks dalam data besar. Aplikasi umum *Deep Learning* yang banyak digunakan adalah pengenalan gambar dan suara. Pada beberapa literatur disebutkan bahwa *Deep Learning* adalah subset dari *Machine Learning*, dan *Machine Learning* adalah subset dari Kecerdasan Artifisial. Gambar 5.3 berikut menunjukkan ilustrasi keterkaitan Kecerdasan Artifisial, *Machine Learning*, dan *Deep Learning*.

Gambar 5.3. Kecerdasan Buatan (Artificial Intelligence)

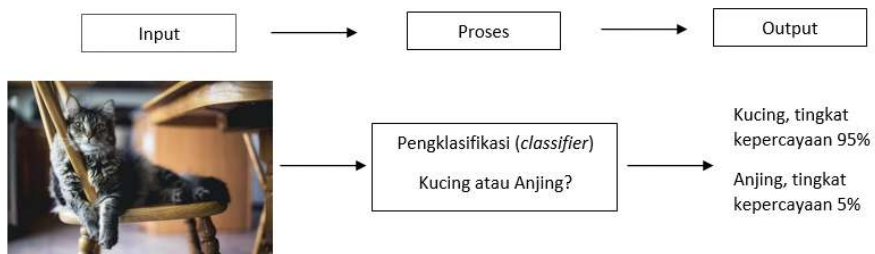
Machine Learning adalah bagian dari bidang ilmu kecerdasan artifisial yang mempelajari cara membuat mesin atau sistem yang memiliki kecerdasan dan menyerupai manusia yang mampu belajar dan memecahkan masalah. Beberapa contoh dari *machine learning* adalah *search engine* pada peramban, (contoh: Google, Bing), sosial media yang memiliki kemampuan memberi saran kepada pengguna yang biasa disebut recommendation system (contoh: Youtube, Netflix, E-Commerce), mobil otonom

(contoh: Tesla), game dengan pengambilan keputusan otomatis (contoh: *strategic game*).

Machine learning diharapkan menjadi sistem yang mampu belajar terus menerus. Dengan semakin banyak data yang dipelajari, sistem akan menjadi semakin pintar.

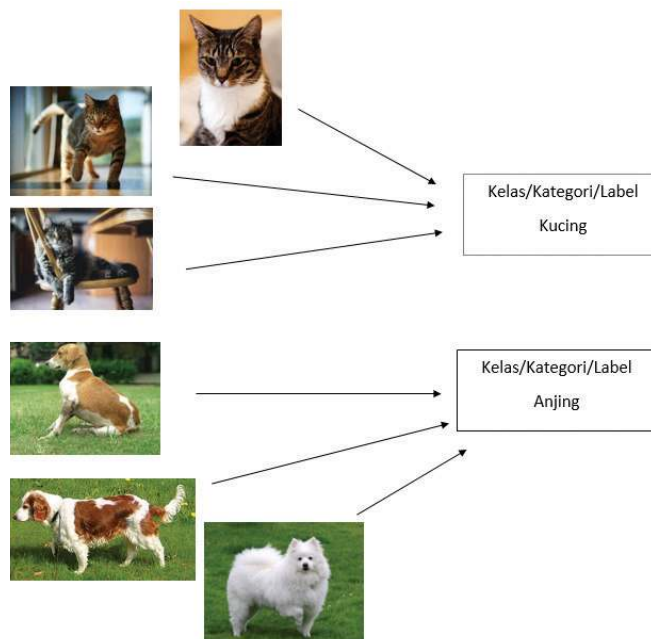
Bagaimana cara kerja machine learning untuk klasifikasi gambar (*image*)?

Klasifikasi gambar adalah salah satu fitur penting pada *machine learning*, sebagai contoh pada mobil otonom yang dapat bergerak tanpa sopir, mobil ini harus mampu dengan cepat untuk menginterpretasikan dan mengklasifikasi sebuah objek yang dilihat dari kamera. Mobil harus menentukan apakah objek tersebut kendaraan lain, pejalan kaki (*pedestrian*) atau rambu lalu lintas. Hal ini sangat penting bagi mobil otonom karena sangat berpengaruh pada gerak jalan mobil.

Salah satu *library/extension* perkakas machine learning di MIT App Inventor yang mampu mengklasifikasikan gambar adalah LookExtension. Library ini dapat menerima gambar yang diambil dari kamera sebagai input dan mampu mengklasifikasi gambar input tersebut menjadi output yang disajikan dalam bentuk teks / tulisan. Contoh pada gambar 5.4 adalah klasifikasi sebuah gambar apakah kelas/kategorinya adalah kucing atau anjing.

Gambar 5.4. Ilustrasi Proses Klasifikasi Gambar

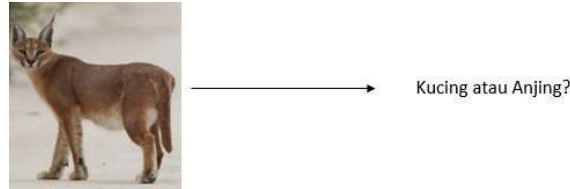
Untuk mendapatkan output klasifikasi berupa teks "Kucing (tingkat kepercayaan 95%)" atau "Anjing (tingkat kepercayaan 5%)" terjadi proses penghitungan/komputasi yang cukup

tinggi dan kompleks. Proses komputasi akan menghasilkan kesimpulan sebuah gambar adalah kucing atau anjing.

Agar dapat menghasilkan kesimpulan yang mampu mengklasifikasikan gambar, sistem pada awalnya **dilatih** untuk dapat mengklasifikasikan gambar (*image*) tertentu. Sebagai contoh, jika kalian ingin mengembangkan sistem untuk dapat mengenali dan mengklasifikasi gambar kucing atau anjing, maka kalian harus **melatih** sistem dengan memberikan banyak gambar kucing dan memberi nama kategori (label) kucing dan memberikan banyak gambar anjing serta memberi label atau kategori anjing juga agar sistem dapat mengingat dan mengenalinya. Memberikan kategori atau label sangat penting bagi sistem, sehingga ketika sistem diberikan input gambar baru, sistem akan dapat menentukan apakah gambar tersebut lebih mirip gambar kucing atau anjing.

Gambar 5.5. Gambar kucing dan anjing dan kelasnya

Dengan contoh gambar kucing dan anjing yang cukup, program akan terlatih untuk menentukan/mengklasifikasi suatu gambar adalah anjing atau kucing. Secara umum, semakin banyak gambar

yang dilatihkan untuk tiap tiap kelas, sistem akan menjadi semakin baik dan andal ketika mengklasifikasikan gambar baru.

Ketika sistem telah dilatih dengan gambar yang cukup, selanjutnya sistem dapat diuji dengan memberikan gambar baru yang tidak diberikan pada saat pelatihan (*training*).

Gambar 5.6. Pengujian dengan gambar baru

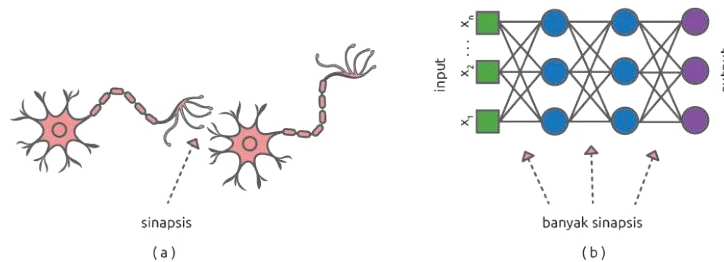
Menurut kalian apa hasil klasifikasi dari gambar diatas? Jika hasil klasifikasinya keliru maka sistem *machine learning* kita dapat dilatih dan diuji kembali dengan gambar baru tadi, sama seperti manusia yang terus menerus belajar. Pelatihan/pembelajaran yang terus menerus membuat sistem kita semakin pintar.

Tapi kita harus berhati-hati karena sistem kita hanya dirancang untuk hanya mampu mengklasifikasikan gambar yang telah kita latihkan, dalam hal ini adalah kucing atau anjing. Sebuah gambar yang sangat berbeda bisa jadi akan diklasifikasi sebagai kucing atau anjing. Sebagai contoh gambar kuda berikut, bisa saja terklasifikasi sebagai kucing atau anjing, yang merupakan klasifikasi yang salah. Jadi karena sistem pengklasifikasi kita hanya bisa membedakan dua kelas/kategori/label maka gambar berbeda akan diklasifikasi pada kedua kelas tersebut. Kelas dapat dikembangkan/ditambah untuk kelas/kategori yang lain dengan proses pelatihan dan pengujian kembali.

Gambar 5.7. Pengujian dengan gambar yang sangat berbeda dari kelas

Salah satu algoritma penting untuk klasifikasi gambar pada *Machine Learning* adalah ***Artificial Neural Network*** (Jaringan Saraf Buatan). *Artificial Neural Network* (ANN) diinspirasi dari cara kerja otak manusia yang terdiri dari kumpulan neuron. Pada sub berikut kalian akan belajar dasar dari algoritma ANN.

Artificial Neural Network

ANN adalah algoritma machine learning yang digunakan pada *library/extension* LookExtension MIT di IDE App Inventor. LookExtension akan kalian eksplorasi pada aktivitias selanjutnya. ANN memiliki cara untuk mepresentasikan pengetahuan dalam kumpulan *node* yang diilhami dari kumpulan *neuron* pada otak manusia yang saling tersambung. Hubungan antar *node* digambarkan dalam bentuk garis yang diilhami oleh sinapsis pada otak manusia. Pengetahuan pada *node* dan sinapsis tercipta melalui runtunan proses komputasi random, spesifik, dan saling terkait dengan *node* lain. Ilustrasi ANN sederhana tampak pada gambar berikut, node digambarkan dalam bentuk bulatan dan garis (hubungan antar node) digambarkan dengan garis panah.

Gambar 5.8. (a) Jaringan otak manusia, (b) *Artificial Neural Network*

Node input pada lapisan input menerima input, maka gambar akan direpresentasikan dalam bentuk data di node input, node berikutnya akan melakukan komputasi untuk menentukan kelas apa dari gambar pada input tersebut. Garis panah memiliki bobot/*weight* berbeda yang tampak dengan tebal tipisnya garis panah.

Proses komputasi yang terjadi dengan melibatkan bobot garis panah dan *node* terus berlanjut lapisan berikutnya yang akan dibandingkan dengan gambar sesuai label. Ada proses umpan balik (*feedback*) yang terjadi yang terus menerus, perbedaan pembandingan akan memicu perubahan bobot pada garis panah sehingga membentuk konfigurasi yang optimal. Umpan balik akan berhenti saat konfigurasi telah dianggap optimal.

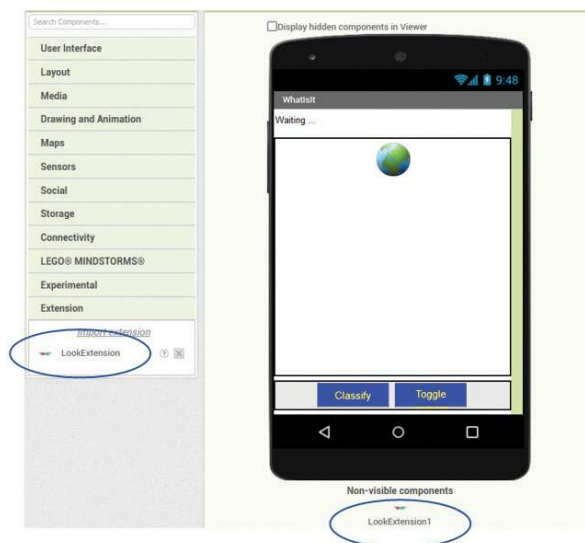
Proses pelatihan adalah proses untuk memperbaharui bobot pada garis panah menuju kondisi optimal yang terbaik untuk pengklasifikasian pola. Pelatihan dianggap cukup, jika pengklasifikasian telah mampu mengklasifikasinya gambar sesuai dengan labelnya, bahkan juga jika gambar adalah gambar baru.

Proses pelatihan adalah komputasi yang cukup kompleks seperti bagaimana menghitung error pada saat pembandingan, pembaharuan bobot, dan termasuk bagaimana gambar untuk pelatihan telah mewakili keseluruhan dari gambar yang akan diklasifikasi. Hal ini terus menjadi topik riset di *Machine Learning* dan ANN sampai saat ini.

Pada ANN dikenal dua jenis pelatihan yang disebut *supervised* dan *unsupervised learning*. *Supervised learning* adalah cara umum yang digunakan untuk klasifikasi gambar, dimana gambar input dan label telah diketahui. Contoh pada klasifikasi gambar dengan kelas/kategori/label "kucing" dan "anjing", adalah salah satu contoh *supervised learning*.

Jenis pelatihan *unsupervised learning* menggunakan cara dimana input tersedia, namun kelas/kategori/label belum diketahui. Pelatihan jenis ini biasanya digunakan untuk mencari pola baru, misalnya dari data perjalanan para turis, data medis yang akan dicari pola baru yang belum diketahui sebelumnya.

LookExtension

Ekstensi LookExtension adalah *library* Neural Network dengan jenis mobilenet yang secara khusus dirancang untuk mengenali gambar dengan menggunakan ponsel. Mobilenet sebenarnya telah dilatih untuk mengenali 999 kelas, dengan jutaan gambar. Kelas gambar tersebut dapat diakses di tautan https://github.com/mit-cml/appinventor-extensions/blob/extension/look/appinventor/docs/reference/components/classes.txt termasuk dengan labelnya. Ekstensi ini bukan merupakan kode inti dari App Inventor namun dapat digunakan dengan App Inventor dengan melakukan impor ekstensi. Tampilan LookExtension pada proyek "Whatisit" tampak pada gambar berikut:

Gambar 5.9. LookExtension pada proyek Whatisit

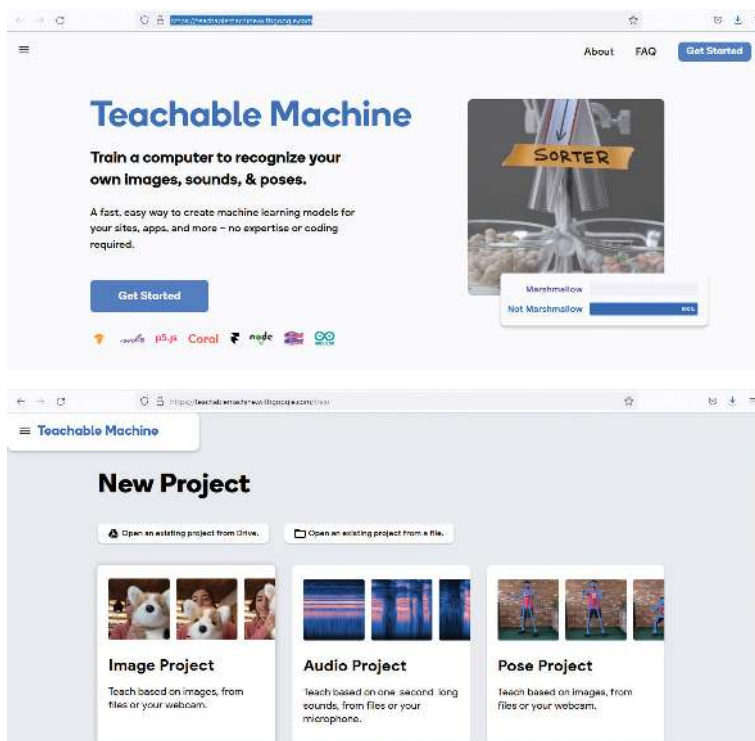
Ayo Kembangkan!

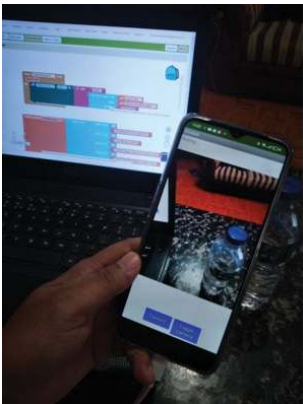
Aktivitas Kelompok

Aktivitas PLB-AI-K11-03: Image Classifier dengan App Inventor

Aktivitas ini akan mengajak kalian untuk belajar dasar *machine learning* sebagai bagian dari kecerdasan artifisial dengan

membuat sendiri aplikasi mobile yang mampu menerapkan *machine learning* untuk mengklasifikasi gambar.

Sebagai pemanasan kalian diajak untuk bermain main terlebih dahulu dengan sistem *machine learning* dari Google yang dapat diakses pada situs berikut: https://teachablemachine.withgoogle.com/. "Teachable Machine" ini dapat mengklasifikasikan gambar (*image*), suara (*audio*), dan pose. Berikut tangkapan layar dari Teachable Machine. Kalian akan dituntun oleh guru untuk melakukan pemanasan pengenalan gambar (*image recognition*) dengan *teachable machine* ini.

Setelah pemanasan selesai, selanjutnya kalian akan beraktivitas untuk mengembangkan aplikasi mobile yang mirip dengan "*Teachable Machine*" dari Google tersebut, namun pengembangan kali ini tidak dilakukan dari nol (*from scratch*) tapi dari proyek yang belum selesai.

Kebutuhan Alat:

a. Komputer yang terkoneksi dengan internet, ponsel/tablet dengan sistem operasi Android/iOS yang memiliki kamera.
b. Komputer juga harus terpasang perangkat lunak MIT AI2 Companion. Jika ponsel tidak tersedia dapat digunakan emulator.

Prasyarat:

Kalian sebagai peserta didik harus telah memahami pemrograman dengan Scratch yang dipelajari di SMP.

Deskripsi Produk:

Kalian akan belajar membuat perangkat lunak berbasis mobile yang dapat mengklasifikasikan gambar dengan LookExtension. Perangkat lunak berfungsi dengan alur, jika pengguna memotret sebuah objek menggunakan kamera ponsel/tablet, maka informasi kelas/kategori dari objek yang dipotret tertampil di layar ponsel. Kelas/kategori ditentukan dengan tingkat kepercayaan tertentu. Kalian akan mengembangkan aplikasi mobile yang memiliki antarmuka sebagai berikut:

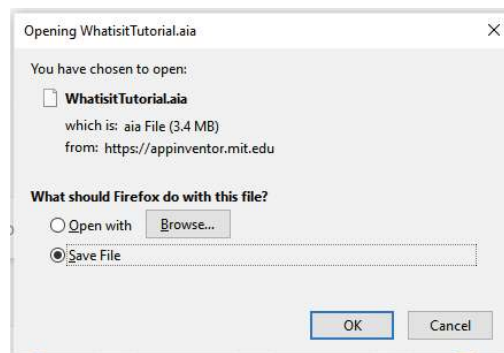
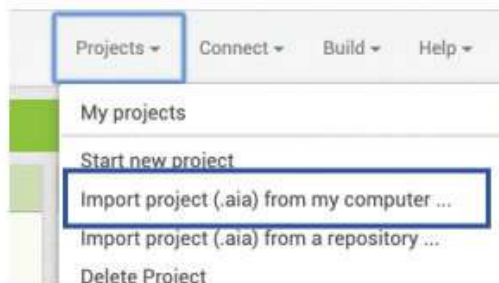
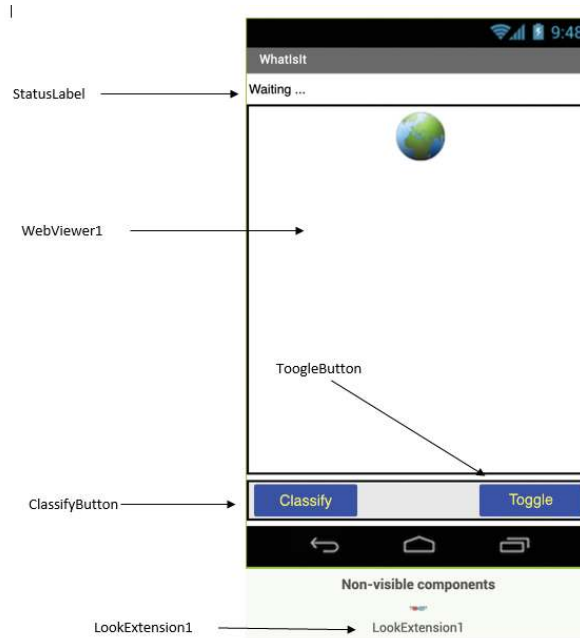
Spesifikasi:

- **Input**: Pengguna mengarahkan kamera ponsel pada objek tertentu dan menekan `Classify` pada aplikasi
- **Proses**: Sistem akan mengklasifikasi objek
- **Output**: Sistem menampilkan teks klasifikasi objek pada layar ponsel dengan tingkat keakuratannya

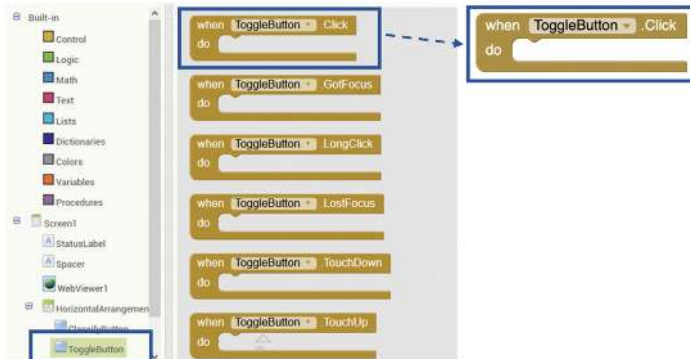
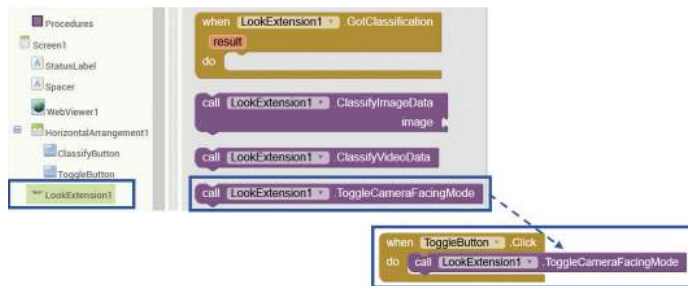
Sistem juga dilengkapi dengan fitur untuk menghidupkan kamera ponsel, dan mengubah kamera yang aktif, apakah kamera depan atau belakang.

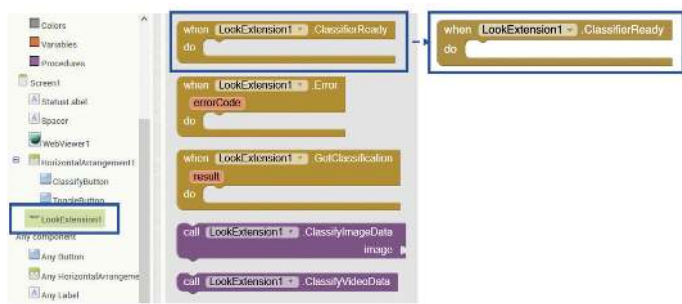
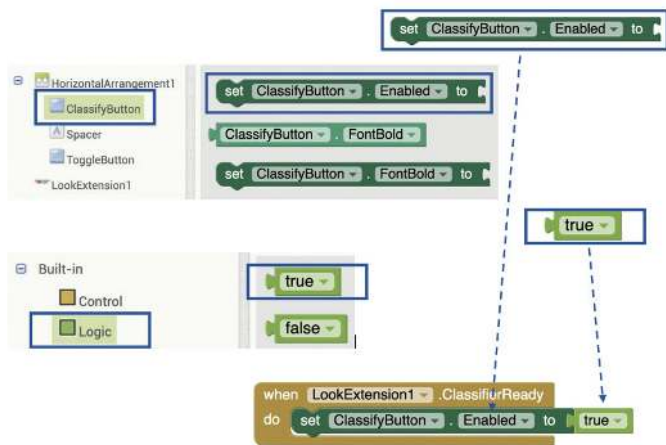
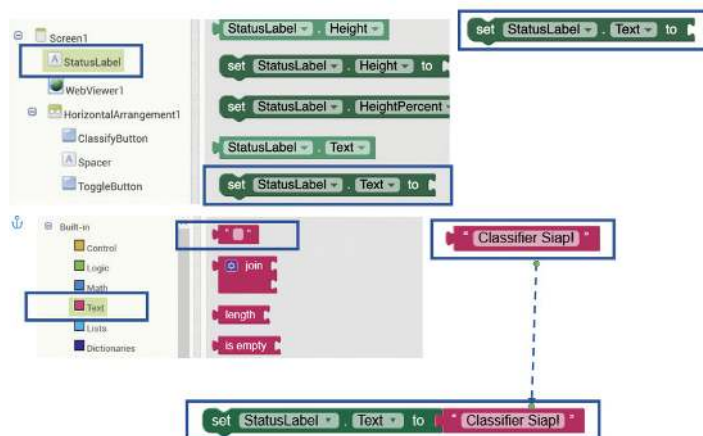
Langkah:

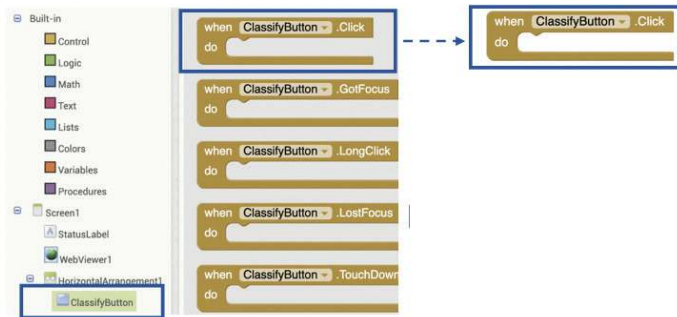
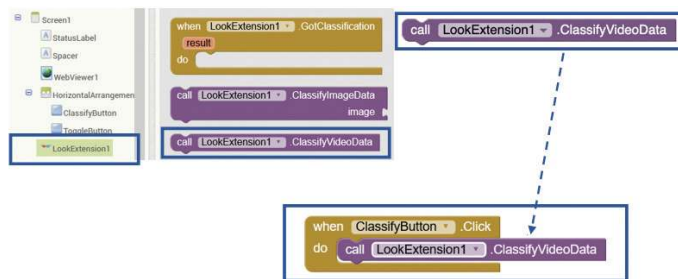
1. Persiapan:

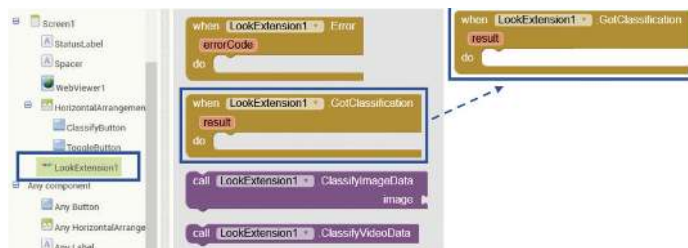
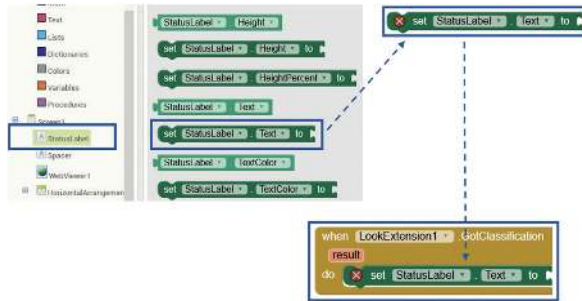
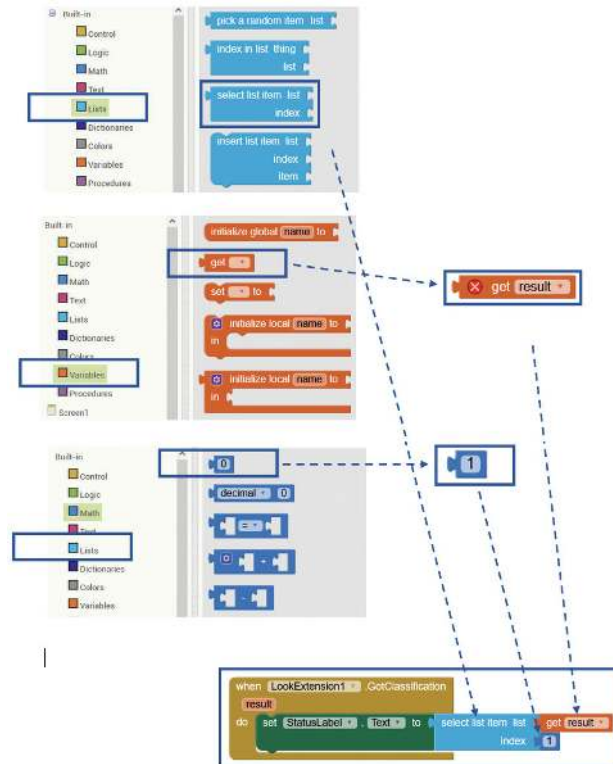
a. *Download* template aplikasi *machine learning* di https://appinventor.mit.edu/assets/files/WhatisitTutorial.aia

b. *Import file* hasil unduhan ke dalam IDE App Inventor sebagai proyek, dengan memilih menu Import project (.aia) *from my computer*.

Tampilan dari proyek hasil impor tampak pada gambar berikut, perhatikan tampilan antarmuka penggunanya, dan cari tahu apa kegunaan masing-masing komponen:

c. Eksplorasi kode yang tersedia, dengan melihat blok yang telah terdefinisi di editor blok. Ada beberapa blok telah tersedia, yaitu:

2. **Pengkodean program** (Modifikasi program): Modifikasi program dengan menambahkan kode blok sebagai berikut:

a. Menambahkan kode button `ToggleButtton` untuk Toggle Camera, dengan blok `when ToggleButton.Click`

b. Isi `ToggleButton.Click` dengan blok `LookExtension1.ToggleCameraFacingDown`

c. Mempersiapkan fungsi klasifikasi di ekstensi `LookExtension`: `LookExtension` sebagai *library* Pengklasifikasi Gambar berhubungan dengan beberapa komponen agar fungsi *classifier* dapat berfungsi dengan baik. Langkah untuk mempersiapkan `LookExtension` adalah:

1) *Drag and drop* fungsi `ClassifierReady` dari komponen `LookExtension1` ke kolom `Viewer`

2) Isi `LookExtension1.ClassifierReady` dengan set `ClassifyButton.Enabled` dengan `True`

d. Setelah langkah c, `LookExtension` telah siap digunakan. Langkah berikutnya mengeset `StatusLabel` dengan teks “`Classifier Siap!`”.

e. Selanjutnya tambahkan blok kode untuk tombol `ClassifyButton`, yaitu:

1) *Drag and drop* blok when `ClassifyButton.Click`

2) Isikan blok `Call LookExtension1.ClassifyVideoData` ke blok `when ClassifyButton.Click`

f. Setelah fungsi klasifikasi pada `LookExtension` selesai di set, selanjutnya tambahkan kode untuk menampilkan hasil klasifikasi pada `StatusLabel`.

1) *Drag and drop* blok `LookExtension.GotClassification`

2) Masukkan `StatusLabel.Text` ke dalam blok `LookExtension.GotClassification`

3) Isi `StatusLabel.Text` dengan hasil dari klasifikasi yang berupa list of list, dengan format `[[klasifikasi1,akurasi1],[klasifikasi2,akurasi2],…,[klasifikasi10,akurasi10]]`, dimana klasifikasi1 adalah klasifikasi dengan akurasi terbaik. Tarik select list item index dari komponen built-in (*lists*), dan masukkan ke `StatusLabel.Text`, seperti pada gambar berikut:

g. Langkah berikutnya adalah pengaturan blok tambahan. Blok tambahan digunakan untuk memberikan informasi ke pengguna jika proses klasifikasi terdapat kesalahan. Blok tambahan telah tersedia pada template kode, yang seperti tampak pada gambar:

3. **Pengujian Sistem**: Setelah kode selesai disusun, uji aplikasi untuk mengklasifikasi beberapa objek dengan mengambil gambar dari beberapa sudut pengambilan seperti dari depan, samping, dan belakang.

4. Selanjutnya, setelah pengujian selesai. Kalian isilah tabel pengujian berikut:

Tabel Pengujian Aplikasi Image Classifier

Diisi saat perencanaan			Diisi setelah Pengujian		
No.	Fitur	Dikerjakan Oleh	Sesuai dengan spesifikasi		Keterangan Hasil Pengujian
			YA	TIDAK	
1.	a. Mengaktifkan kamera ponsel:				
	Kamera Depan				
	Kamera Belakang				

Diisi saat perencanaan			Diisi setelah Pengujian		
No.	Fitur	Dikerjakan Oleh	Sesuai dengan spesifikasi		Keterangan Hasil Pengujian
			YA	TIDAK	
2.	b. Mengenali objek tertentu:				
	Mouse - pengambilan gambar menghadap ke depan				
	Mouse - pengambilan gambar menghadap samping				
	Botol Air - pengambilan gambar menghadap ke depan				
	(Gunakan objek lainnya)				

Ide Pengembangan:

1. Aplikasi dapat dikembangkan dengan menggabungkan aktivitas PLB-AI-K11-01 dengan PLB-AI-K11-03 yang membuat hasil klasifikasi dari classifier menjadi suara.
2. Aplikasi dapat juga dikembangkan untuk menampilkan hasil klasifikasi untuk dua item terbaik. Jadi tidak hanya satu item hasil klasifikasi yang tertampil di layar tapi dua item.
3. Bagaimana kalau aktivitas PLB-AI-K11-03 dikembangkan untuk mengenali suara? Atau gambar saja (bukan video) seperti kode diatas?
4. Apa ide pengembangan aplikasi kalian?

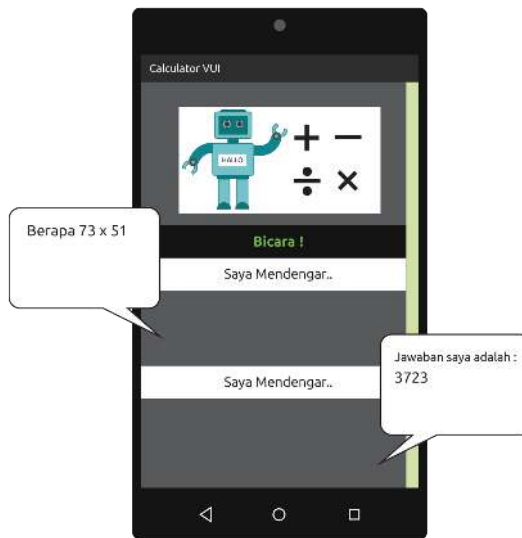
Ayo Kerjakan!

Aktivitas Kelompok

Aktivitas PLB-AI-K11-04: Kalkulator dengan Suara

Pernahkan kalian menggunakan pencarian dengan Google dengan suara? Atau bertanya dengan menggunakan kepada Alexa dan Siri? Bagaimana perangkat lunak tersebut menginterpretasi apa yang kita ucapkan? Dan bagaimana aplikasi tersebut merespon permintaan kita?

Tujuan dari pengembangan proyek kecerdasan artifisial adalah untuk memberi pemahaman tentang dasar-dasar antarmuka pengguna berbasis suara (VUI) serta proses perancangan sistem kecerdasan artifisial sederhana yang dapat memahami pengguna dalam pertanyaan dan tanggapan perhitungan yang dinyatakan secara lisan dengan tepat. Sistem kecerdasan artifisial yang digerakkan oleh suara seperti ini dapat berguna dalam berbagai konteks seperti saat merancang teknologi bantu untuk penyandang disabilitas visual dan orang tua. Misalnya, pengguna tunanetra dapat menggunakan kalkulator suara untuk melakukan perhitungan matematis secara verbal tanpa harus mengetikkan semua detail perhitungan.

Aktivitas ini adalah aktivitas pengembangan perangkat lunak berbasis *Artificial Intelligence* menggunakan library yang telah ada di App Inventor dengan contoh desain layar pada gambar berikut.

Pada projek ini kalian ditantang untuk membuat aplikasi yang mampu menggunakan *Voice User Interface* untuk melakukan penghitungan aritmatika sederhana. Aplikasi yang dikembangkan mampu untuk:

a. Melakukan interpretasi suara yang memerintahkan operasi aritmatika yaitu:

- Penjumlahan (+)
- Pengurangan (-)
- Perkalian (x)
- Pembagian (:)

b. Menghitung operasi aritmatika tersebut dan menampilkan hasilnya di layar serta memperdengarkan dalam bentuk suara.

Catatan: Proyek ini tidak dapat menggunakan emulator pada pengujian karena aplikasi tergantung pada kemampuan pengenalan suara pada ponsel. Ponsel juga harus memiliki kemampuan pengenalan suara agar proyek dapat berfungsi.

Gunakan **Lembar Kerja Peserta Didik 2** berikut:

Peran	Penanggung Jawab
Analis Program: a. Deskripsi Produk b. Spesifikasi Aplikasi c. Kebutuhan resource: file, alat, dll	
Perancang User Interface (UI)	
Pemrogram Kode	
Penguji Program	
Pemapar Presentasi	

Spesifikasi (Deskripsi Produk, Fungsionalitas Aplikasi, Kebutuhan *Resource*)

Rancangan *User Interface* (UI)

Kode Program

Pengujian

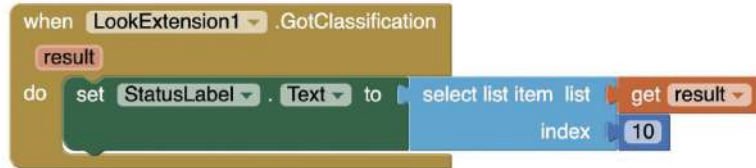
Tabel 5.3 Format LKPD-02 Aktivitas PLB-AI-K11-04

Diisi saat perencanaan			Diisi setelah Pengujian		
No.	**Fitur**	**Dikerjakan Oleh**	**Sesuai dengan spesifikasi**		**Keterangan Hasil Pengujian**
			YA	**TIDAK**	

Uji Kompetensi

Soal Pilihan Ganda

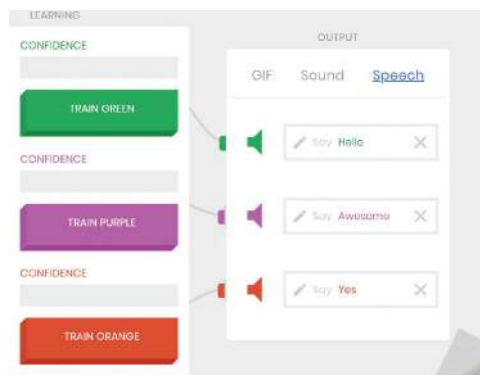
1. Jika pada LookExtension1 kalian mengubah nilai kode index dari 1 ke 10. Apa yang akan terjadi ?

```
when LookExtension1 .GotClassification
  result
do  set StatusLabel . Text to  select list item  list  get result
                                                 index  10
```

a. Menampilkan 10 klasifikasi terbaik

b. Terjadi pengulangan 10 kali untuk mendapatkan klasifikasi terbaik

c. Menampilkan pesan kesalahan

d. Tidak terjadi apa apa

2. Pada Teachable Machine, jika anda melatih sasando dengan warna hijau, angklung dengan warna purple, dan gamelan dengan warna orange. Dan kalian menguji dengan gambar gamelan. Suara apa yang akan terdengar?

a. Hello

b. Awesome

c. Yes

d. Tidak keluar suara apapun

Ayo Renungkan

Setelah selesai melakukan aktivitas tersebut. Jawablah pertanyaan berikut ini dalam Lembar Refleksi pada Buku Kerja, dan jangan lupa mencatat kegiatan dalam Jurnal Peserta Didik.

1. Apakah kalian telah memahami cara mengembangkan aplikasi berbasis mobile dengan App Inventor ditambah dengan extension LookExtension?
2. Apakah kalian telah memahami cara pengujian artefak komputasional, berupa aplikasi mobile?
3. Kesulitan apa yang kalian temukan pada materi ini?
4. Ide aplikasi apa yang kalian pikirkan untuk pengembangan aplikasi yang telah kalian kembangkan dengan pustaka Kecerdasan Artifisial?

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
REPUBLIK INDONESIA, 2021
Informatika untuk SMA Kelas XI
Penulis : Paulina H. Prima Rosa
ISBN: 978-602-244-861-7

Proyek Analisis Data: "Hutanku Dulu, Kini, dan yang Akan Datang"

Tujuan Pembelajaran

Setelah mempelajari bab ini kalian diharapkan mampu:

1. Memiliki budaya kerja masyarakat digital dalam tim dengan rekan-rekan yang memiliki berbagai macam latar belakang.
2. Berkolaborasi untuk melaksanakan tugas dengan tema komputasi.
3. Mengenali dan mendefinisikan persoalan lingkungan hidup yang pemecahannya dapat didukung dengan sistem komputasi, khususnya analisis data.
4. Mengembangkan dan menggunakan abstraksi untuk memodelkan masalah melakukan prediksi.

5. Mengembangkan artefak komputasional dengan melakukan analisis data, serta memvisualisasikan hasilnya untuk menunjang model komputasi yang dibutuhkan di pelajaran lain.
6. Mengembangkan rencana analisis data dan visualisasinya dan mendokumentasikan hasilnya.
7. Mempresentasikan hasil analisis data secara lisan dan tertulis dalam bentuk infografis, peta pikiran, serta poster dengan memperhatikan hak kekayaan intelektual dan hak pribadi.

Pertanyaan Pemantik

Saat ini kurang lebih 31% permukaan Bumi tertutup oleh hutan. Persentase hutan yang tersisa ini hanya sepertiga dari keadaan sebelum manusia melakukan perluasan pertanian. Kini, setiap menit rata-rata 2400 pohon ditebang. Menurut kalian, apa yang akan terjadi di masa mendatang bila kejadian ini terus berlanjut? Apa dampaknya bagi manusia dan makhluk hidup lainnya?

Peta Konsep

Gambar 6.1 Peta Konsep Proyek Analisis Data

Praktik Lintas Bidang (PLB) dalam bab 6 ini merupakan sebuah kegiatan untuk menganalisis data, memproses, dan memvisualisasi hasil analisis dalam bentuk infografis. Kegiatan ini berbentuk proyek yang dikerjakan dalam jangka waktu 4 minggu secara berkelompok. Setiap kelompok akan diminta untuk mencari data yang mencerminkan fenomena yang terjadi di lingkungan sekitar untuk dianalisis. Kalian akan saling berinteraksi dalam kelompok untuk mendiskusikan fenomena yang terjadi di lingkungan kalian dan memaknainya. Kalian akan mendapatkan pengarahan dari guru tentang proyek PLB ini terlebih dahulu sebagai orientasi. Selanjutnya, pengetahuan dan keterampilan tentang analisis data yang pernah kalian dapatkan dari kelas VII hingga Kelas X dapat kalian manfaatkan untuk menyelesaikan proyek ini. Proyek ini akan melatih

setiap anggota tim untuk mengelola setiap langkah pekerjaan. Produk dari PLB ini adalah aneka dokumentasi proyek dan hasil analisis data yang dituangkan dalam bentuk infografis, poster, serta peta pikiran.

Apersepsi

Hutan memegang peran penting dalam kehidupan manusia karena dapat menyerap CO2. Menurut data dari Kementerian Lingkungan Hidup dan Kehutanan, luas hutan di Indonesia pada tahun 2020 adalah 95.6 juta hektar. Oleh karenanya, Indonesia sering disebut sebagai paru-paru dunia. Sayangnya kian hari kian banyak hutan yang beralih fungsi menjadi lahan tidak berhutan secara permanen. Hal ini sering disebut sebagai deforestasi. Apa akibat deforestasi bagi manusia? Bagaimana peta deforestasi hutan-hutan di Indonesia? Data apa saja yang yang perlu dan dapat kita peroleh secara gratis serta terbuka untuk umum sehingga dapat kita analisis lebih jauh?

Kata Kunci

Deforestasi, analisis data, visualisasi, klasifikasi, prediksi

A. Pengarahan Guru dan Observasi

Sebagai negara tropis dengan hutan terluas kesembilan di dunia, Indonesia memiliki kekayaan alam yang luar biasa. Selain menjadi paru-paru dunia, di dalam hutan tropis terdapat keanekaragaman hayati yang sangat kaya. Berbagai spesies tumbuhan maupun hewan banyak kita temukan di hutan tropis Indonesia. Beberapa diantaranya bahkan hanya ada di Indonesia.

Sekalipun demikian, aktivitas manusia yang ekspansif mengakibatkan beralihnya hutan menjadi perkebunan maupun hunian. Deforestasi ini pasti mengancam kelestarian makhluk hidup di dalam hutan tropis tersebut. Bayangkanlah

berapa banyak makhluk hidup yang hanya ada di hutan tropis Indonesia yang akan punah dengan lenyapnya hutan.

Dalam bab ini, kalian akan belajar di dalam kelompok untuk mencermati data yang terkait dengan deforestasi, menganalisis dan menyajikannya secara visual mempergunakan pengetahuan tentang analisis data yang telah kalian dapatkan di kelas-kelas sebelumnya.

1. Pengarahan Guru

Kalian akan mendapatkan penjelasan dari guru tentang pelaksanaan PLB ini. Sebagai permulaan, kalian wajib mencatat berbagai macam informasi yang disampaikan oleh guru. Kalian dapat mencatat penjelasan guru dalam Buku Kerja menggunakan contoh format yang diberikan di bawah ini, yang tentu saja boleh kalian sesuaikan dengan gaya penulisan kalian masing-masing.

Tabel 6.1 Format Catatan Informasi Penjelasan Guru

Hari / Tanggal Kegiatan	
Narasumber	
Catatan	**Deskripsi Kegiatan:**
	Yang perlu disiapkan:
	Timeline / rencana jadwal kegiatan PLB:

2. Observasi

Untuk memahami persoalan tentang deforestasi, kalian dapat menyimak beberapa video dari sumber-sumber yang disebutkan di bawah ini, kemudian mencatat beberapa hal yang dapat kalian temukan dengan menggunakan beberapa pertanyaan panduan dalam PLB-AD-LKPD-01. Video dapat kalian tonton bersama-sama dalam kelas menggunakan perangkat LCD, atau kalian tonton sendiri-sendiri menggunakan gawai. Jika perangkat-perangkat tersebut tidak tersedia, kalian bisa membaca artikel tentang deforetasi yang dibagikan oleh guru kalian.

Berikut ini 3 tautan video yang dapat kalian simak. Jika kalian terkendala memahami bahasa Inggris, maka cobalah bereksplorasi mencari cara untuk memunculkan subtitle berbahasa Indonesia dalam tayangan youtube berikut ini:

1. *What is deforestation:* https://youtu.be/vJnnrpSDWPI
2. *Climate 101*: *Deforestation*: https://youtu.be/Ic-J6hcSKa8
3. *Deforestation Effects on Climate*: https://youtu.be/Nc7f5563azs
4. *Rainforest deforestation and its effects*: https://youtu.be/AVh2DEgpvsM

Ayo Berpikir!

Aktivitas Individual

Aktivitas PLB-AD-K11-01-U: Apa dan Mengapa Deforestasi

PLB-AD-LKPD-01 berikut ini adalah lembar kerja yang berisi beberapa pertanyaan untuk membantu kalian menyarikan dan mengkritisi apa yang kalian simak. Secara individual, tuliskan temuan, pendapat, dan gagasan kalian dalam LKPD ini. Kalian juga bisa membuat pertanyaan tambahan lain dan menjawabnya sendiri dalam bagian yang kosong dari LKPD ini.

Tabel 6.2 PLB-AD-LKPD-01 Pengenalan Deforestasi

No	Pertanyaan	Jawabanku
1.	Apa sesungguhnya makna deforestasi itu?	
2.	Mengapa deforestasi terjadi?	
3.	Di manakah terjadi deforestasi? Apakah hal itu terjadi di Indonesia juga?	
4.	Kapan terjadi deforestasi?	
5.	Apa saja dampak deforestasi?	
6.	Siapa/apa saja yang terkena dampak dari terjadinya deforestasi? Seperti apa dampak pada yang terkena tersebut dan mengapa bisa demikian?	
7.	Apakah deforestasi bisa dihentikan? Mengapa?	
8.	Apa yang bisa kita lakukan untuk mengatasi persoalan deforestasi?	
9.	Apakah ada hubungan deforestasi dengan perubahan iklim? Jika ya, bagaimana hubungannya?	
10.	Menurutmu, apakah kita perlu peduli terhadap persoalan deforestasi? Mengapa?	
11.	Bagaimana bisa disimpulkan bahwa jika laju deforestasi tetap seperti sekarang maka dalam 100 tahun mendatang hutan kita akan hilang? Apa yang dilakukan untuk bisa mengambil kesimpulan seperti itu?	

No	Pertanyaan	Jawabanku
12.	....	
13.	....	
14.	....	
15.	....	

Ayo Berdiskusi!

Aktivitas Berkelompok

Aktivitas PLB-AD-K11-02-U: Visualisasi Persoalan Deforestasi

Diskusikan jawaban kalian dalam kelompok, kemudian bersama kelompok gambarkan sebuah ilustrasi tentang persoalan deforestasi ini mulai dari penyebab serta dampak yang diakibatkannya pada berbagai hal yang telah kalian temukan dan catat dalam PLB-AD-LKPD-01 di atas. Masing-masing kelompok boleh berkreasi membuat gambar yang diimajinasikan, semenarik mungkin.

Deskripsi Proyek

Setelah kalian mencoba mendalami persoalan deforestasi dan menggambarkan berbagai unsur yang saling terkait dalam persoalan tersebut, guru akan meminta kalian untuk mencari data yang berkaitan dengan deforestasi dari setiap provinsi di Indonesia. Sebagai contoh, kalian dapat mencari data tentang luasan lahan yang mengalami deforestasi setiap tahunnya, data tentang jumlah satwa di hutan yang terancam

punah, data tentang suhu udara, data tentang banjir, dan sebagainya. Data yang kalian kumpulkan minimal berisi nama subjek pengamatan dan besarannya untuk kurun waktu minimal 5 tahun. Yang dimaksud dengan subjek pengamatan adalah subjek yang diamati, misalnya nama provinsi/hewan/tanaman/dsb. Yang disebut dengan besaran misalnya suhu/jumlah hewan/jumlah tanaman. Data itu akan menjadi bahan untuk dianalisis lebih lanjut dalam unit ini. Kalian bisa mencari datanya dari internet atau sumber lain yang bisa kalian dapatkan.

Gambar 6.1 berikut ini adalah salah satu contoh penggalan data tentang deforestasi yang diambil dari https://www.bps.go.id/. Kalian bisa mencari data terbuka lain dari internet tentang deforestasi dengan menggunakan kata kunci yang tepat misalnya "data deforestasi indonesia", "*deforestation open data*", dan sebagainya.

Angka Deforestasi Netto Indonesia Di Dalam Dan Di Luar Kawasan Hutan Tahun 2013-2019 (Ha/Th)

	Provinsi	Total deforestasi					
		2013-2014	2014-2015	2015-2016	2016-2017	2017-2018	2018-2019
5	Aceh	7.648,1	3.231,5	23.071,6	15.515,7	7.502,2	11.608,0
6	Sumatera Utara	6.140,5	20.665,7	9.714,3	22.549,6	7.319,1	12.402,4
7	Sumatera Barat	5.056,0	10.498,9	7.817,8	8.680,0	5.514,0	9.324,8
8	Riau	201.939,4	135.530,7	24.190,9	6.981,6	43.714,8	142.011,1
9	Jambi	-9.941,5	16.893,1	24.821,4	32.871,3	9.235,6	27.382,0
10	Sumatera Selatan	3.527,2	290.777,0	4.294,2	22.286,6	3.741,0	60.655,1
11	Bengkulu	12.069,4	4.133,1	1.798,8	5.091,5	7.706,4	2.160,6
12	Lampung	163,9	16.700,2	1.213,6	4.420,3	1.358,5	625,4
13	Kepulauan Bangka Belitung	1.699,4	19.292,3	1.692,3	3.181,7	3.745,8	2.154,4
14	Kepulauan Riau	4,6	1.322,0	697,7	5.662,8	-142,5	562,0
15	DKI Jakarta	-	-	-	-0,8	-	-
16	Jawa Barat	-14.880,5	6.060,0	-	-188,0	6.388,5	4.977,4
17	Jawa Tengah	-94,3	5.231,4	-	-	3.115,6	1.904,0
18	DI Yogyakarta	32,8	51,8	-	-270,2	589,8	326,4

Gambar 6.2 Contoh Penggalan Data Deforestasi

Data yang kalian dapatkan selanjutnya akan kalian cermati dan analisis dengan menggunakan beberapa panduan pertanyaan yang akan diuraikan di bagian pelaksanaan proyek. Sebagai produk dari proyek ini, kalian akan menghasilkan:

1. Infografis tentang pengelompokan subjek pengamatan berdasar kategori:
    a. Hijau: kategori baik
    b. Kuning: kategori sedang
    c. Merah: kategori buruk dan butuh prioritas perhatian

    Kriteria pengelompokan boleh kalian tentukan sendiri dengan alasan yang diuraikan secara jelas.

2. Gagasan untuk membuat model prediksi keadaan subjek pengamatan di tahun mendatang berdasar data yang diperoleh. Gagasan dapat kalian tuangkan dalam bentuk:
    a. Peta pikiran (mind map)
    b. Poster layanan masyarakat yang relevan.

    Kalian juga bisa berkreasi menggabungkan infografis hasil produk nomor 1 di atas dalam produk gagasan model prediksi ini.

Dengan mempertimbangkan ketersediaan alat bantu analisis di sekolah maupun rumah kalian, untuk mengerjakan proyek ini kalian boleh memilih salah satu tingkat kompleksitas proyek berikut:

Tabel 6.3 Tingkat Kompleksitas Proyek PLB-AD-K11-02-U

Tingkat Kompleksitas	Alat Bantu Analisis	Luaran (Hasil)
0	Kertas dan alat tulis	• Deskripsi cara analisis yang dilakukan secara manual. • Visualisasi hasil analisis berupa infografis, peta pikiran, poster yang **dibuat menggunakan kertas dan alat tulis lain yang dikehendaki**.

Tingkat Kompleksitas	Alat Bantu Analisis	Luaran (Hasil)
1	Perkakas pengolah lembar kerja (spreadsheet) Kalian bisa melihat kembali pokok bahasan Analisis Data dalam buku Informatika kelas VII dan VIII tentang penggunaan pengolah lembar kerja ini.	• Deskripsi cara analisis yang dilakukan dengan menggunakan perkakas pengolah lembar kerja, • Visualisasi hasil analisis berupa infografis, peta pikiran, poster yang **dibuat menggunakan perkakas pengolah lembar kerja**.
2	Bahasa pemrograman Python Kalian bisa melihat kembali pokok bahasan Analisis Data dalam buku Informatika Kelas X tentang penggunaan python ini.	• Deskripsi cara analisis yang dilakukan dengan menggunakan python, • Visualisasi hasil analisis berupa infografis, peta pikiran, poster yang **dibuat menggunakan python**.

Apa yang akan kalian dapatkan dalam PLB-AD-K11-02-U ini?

Proyek ini memiliki tujuan agar kalian memiliki pengalaman untuk:

1. Berpikir kritis untuk mengidentifikasi suatu masalah, mengenali hubungan sebab akibat dalam suatu masalah, dan keterkaitan berbagai komponen dalam suatu masalah
2. Mencari data yang sesuai dengan topik
3. Memahami makna data yang didapat
4. Mengelompokkan data berdasar kemiripannya
5. Memvisualkan data agar mudah dibaca orang dengan cepat
6. Menemukan gagasan untuk melakukan prediksi berdasar data yang diperoleh

7. Memanfaatkan alat bantu untuk membuat visualisasi dan melakukan prediksi
8. Bekerja sama dalam kelompok

3. Pelaksanaan Proyek

Pada bagian ini kalian akan melaksanakan proyek yang sebelumnya telah disampaikan tujuan proyek ini dari guru kalian pada aktivitas PLB-AD-K11-01-U serta kalian telah melakukan kegiatan observasi proyek pada aktivitas PLB-AD-K11-02-U. Kalian diharapkan dapat melakukan proyek ini dengan sungguh-sungguh, karena selanjutnya kalian akan bermain peran sesuai dengan tugas dan pekerjaannya masing-masing. Mainkanlah peran kalian dengan baik sesuai tugasnya agar hasil proyek menjadi lebih optimal.

Ayo Berdiskusi!

Aktivitas Kelompok

Aktivitas PLB-AD-K11-03-U: Penyusunan Kelompok & Rencana Kerja

Penyusunan Tim/Kelompok

Setelah kalian mendengarkan arahan, kalian akan diminta untuk menyusun tim/kelompok. Bentuklah kelompok yang terdiri atas 4-5 orang. Tuliskan anggota kelompok dalam format seperti Tabel 6.4 berikut. Kalian dapat menggunakan aplikasi pengolah lembar kerja atau aplikasi pengolah kata untuk mencatat/mendokumentasikan hasil kerja kelompok kalian.

Tabel 6.4 Format Catatan Pembagian Tugas Kelompok

NAMA KELOMPOK				
No.	Nama	Peran	No. HP	Email

Pembagian Peran

Setiap orang dalam kelompok akan mendapatkan perannya masing-masing. Peran yang dapat kalian pilih untuk setiap orang anggota kelompok misalnya seperti dalam Tabel 6.5 berikut. Namun kalian juga dapat menentukan pembagian tugas yang berbeda dari contoh ini.

Tabel 6.5 Contoh Peran Anggota Kelompok PLB-AD-K11-03-U

No	Peran	Tugas / Pekerjaan
1.	Project Manager	• Menyusun jadwal kerja • Menyusun pembagian kerja kelompok • Mengkoordinir kelompok • Memonitor jalannya proyek
2.	Pencari Data	• Melakukan pencarian sumber data yang relevan dengan topik • Mengolah dan menyajikan data yang diperoleh menjadi berbentuk tabel yang siap dianalisis lebih lanjut
3.	Analis Data	• Mencermati data yang telah tersaji dalam tabel • Merumuskan cara bagaimana mengelompokkan data ke dalam zona • Merumuskan cara bagaimana melakukan prediksi berdasar data yang ada
4.	Perancang Visual	• Merancang tampilan visual hasil analisis data dalam bentuk infografis, peta pikiran, dan poster • Menyajikan hasil analisis menjadi infografis, peta pikiran, dan poster

Perhatikan bahwa peran-peran tersebut lebih merupakan tanggungjawab utama dari masing-masing pemegang peran. Dalam praktik, **sangat disarankan semua anggota kelompok saling berkolaborasi dan terlibat dalam setiap bagian pekerjaan sehingga memahami dengan utuh bagaimana proyek ini diselesaikan**.

Penyusunan Rencana Kerja

Setelah kalian mendapatkan kelompok dan peran, langkah awal yang perlu dilakukan adalah melakukan penyusunan rencana kerja. Penyusunan rencana kerja dapat menggunakan Gantt chart sederhana dengan format pada Tabel 6.6 berikut. Gantt chart ini telah kalian pelajari dalam Buku Ajar Informatika Kelas X sub bab PLB. Kalian dapat menengok kembali catatan kalian di Kelas X.

Tabel 6.6 Contoh Gantt chart PLB-AD-K11-03-U

No.	Kegiatan	Minggu					PJ	Ket
		1	2	3	4	5		
1.	Pencarian data							
	Diskusi internal kelompok untuk menganalisis data							
2.	Diskusi internal kelompok untuk membuat visualisasi data							
3.	Pembuatan laporan							
4.	Presentasi hasil							

Kolom kegiatan diisi dengan daftar aktivitas yang akan dikerjakan. Kolom minggu merupakan penanda kapan kegiatan tersebut harus dilaksanakan. Kolom penanggung jawab diisi dengan nama anggota kelompok yang merupakan penanggung jawab dari kegiatan tersebut. Berbagai catatan dapat dituliskan di kolom keterangan.

Secara garis besar, kegiatan proyek analisis data ini akan dilakukan berpusat kepada kalian sebagai peserta didik dan berorientasi proyek (project based learning atau PjBL). Kegiatan akan dilaksanakan selama 5 minggu, untuk detail jadwalnya akan ditentukan oleh guru.

Ayo Lakukan!

Aktivitas Kelompok

Aktivitas PLB-AD-K11-04: Pencarian Data

Kegiatan ini merupakan kegiatan untuk mengumpulkan data yang dibutuhkan dari sumber-sumber yang relevan. Luaran (hasil) dari proses ini berupa tabel berisi data yang akan dipergunakan untuk tahapan berikutnya. Tuliskan catatan proses pencarian data dalam tabel Lembar Kerja Peserta Didik 2 (LKPD- 2) berikut ini.

Tabel 6.7 Format PLB-AD-LKPD-02 Catatan Proses Pencarian Data

Pertanyaan	Uraian
1. Bagaimana kelompok menetapkan cara mencari data?	
2. Sumber apa yang dipakai untuk mencari data?	
3. Apa alasan memilih sumber tersebut?	
4. Bagaimana cara melakukan pencarian data?	

Pertanyaan	Uraian
5. Format data yang diperoleh seperti apa?	
6. Apa saja isi dari data yang diperoleh?	
7. Data yang diperoleh mencakup periode kapan?	
8. Kapan data diperoleh?	
9. Siapa saja yang terlibat melakukan pencarian data?	

Ayo Berdiskusi

Aktivitas Kelompok

Aktivitas PLB-AD-K11-05: Analisis Data

Kegiatan ini merupakan kegiatan untuk menganalisis data yang telah didapatkan. Luaran yang diharapkan dari proses ini adalah rancangan langkah-langkah untuk menganalisis. Tuliskan catatan proses analisis data dalam tabel Lembar Kerja Peserta Didik 3 (PLB-AD-LKPD-03-U) berikut ini.

Tabel 6.8 PLB-AD-LKPD-03 Catatan Proses Analisis Data

Pertanyaan	Uraian
1. Level analisis ke berapa yang dipilih oleh kelompok?	
2. Bagaimana kelompok menetapkan cara analisis data dan menetapkan kategorisasinya?	
3. Referensi apa yang dipakai untuk melakukan analisis data?	

Pertanyaan	Uraian
4. Apa alasan kalian memilih referensi tersebut?	
5. Alat bantu apa yang digunakan untuk menganalisis?	
6. Apa saja faktor-faktor penyebab deforestasi yang ditemukan oleh kelompok?	
7. Apa saja dampak deforestasi yang berhasil diidentifikasi oleh kelompok?	
8. Siapa saja yang terlibat dalam melakukan analisis data?	

Ayo Lakukan!

Aktivitas Kelompok

Aktivitas PLB-AD-K11-06: Visualisasi Hasil Analisis

Kegiatan ini merupakan kegiatan untuk merancang dan menampilkan secara visual hasil analisis data. Luaran yang diharapkan dari aktivitas ini adalah infografis berdasar data yang diperoleh. Tuliskan catatan proses visualisasi data dalam PLB-AD-LKPD- 04 berikut ini:

Tabel 6.9 Format PLB-AD-LKPD-04 Catatan Proses Visualisasi Data

Pertanyaan	Uraian
1. Bagaimana cara melakukan visualisasi data?	
2. Alat bantu apa yang dipergunakan untuk visualisasi?	
3. Bagaimana hasil visualisasinya?	

Pertanyaan	Uraian
4. Siapa saja yang terlibat menyusun visualisasi data?	

Ayo Berdiskusi

Aktivitas Kelompok

Aktivitas PLB-AD-K11-07: Perumusan Gagasan Prediksi

Kegiatan ini merupakan kegiatan untuk merumuskan ide bagaimana melakukan prediksi berdasar data yang diperoleh tersebut, misalnya pada tahun berapakah hutan akan habis jika tingkat deforestasi seperti sekarang atau berapakah suhu di suatu provinsi pada tahun mendatang. Luaran yang diharapkan dari aktivitas ini adalah peta pikiran tentang gagasan untuk melakukan prediksi terhadap data yang diperoleh dan perluasan cakupan data tersebut.

Tuliskan catatan proses menemukan gagasan prediksi data dalam PLB-AD-LKPD-05 berikut ini.

Tabel 6.10 Format PLB-AD-LKPD-05 Catatan Proses Perumusan Gagasan Prediksi

Pertanyaan	Uraian
1. Bagaimana kelompok menemukan gagasan untuk melakukan prediksi?	
2. Alat bantu apa yang dipergunakan untuk melakukan prediksi?	
3. Bagaimana hasil peta pikirannya?	
4. Siapa saja yang terlibat dalam menyusun peta pikiran?	

Ayo Lakukan!

Aktivitas Kelompok

Aktivitas PLB-AD-K11-08: Pembuatan Poster

Sebagai wujud aksi menyikapi persoalan deforestasi yang telah kalian kaji dalam bab ini, berkreasilah dalam kelompok untuk merepresentasikan kepedulian kalian dengan berkreasi membuat poster tentang deforestasi. Poster ini dapat kalian manfaatkan untuk mengedukasi dan meningkatkan kesadaran orang lain tentang masalah deforestasi. Kalian bisa menyajikan poster tersebut secara digital maupun tercetak, yang memuat infografis, hasil analisis, maupun gagasan prediksi yang telah kalian diskusikan.

Luaran yang diharapkan: poster digital atau cetak untuk meningkatkan kesadaran orang tentang masalah deforestasi.

Ayo Lakukan!

Aktivitas Kelompok

Aktivitas PLB-AD-K11-09: Pembuatan Laporan

Sebagai dokumentasi seluruh proses yang telah kalian jalani dalam bab ini, dalam kelompok kalian perlu menyusun laporan akhir yang berisi tentang:

1. Deskripsi umum proyek
2. Data yang diperoleh
3. Cara analisis
4. Luaran yang dihasilkan
5. Saran
6. Lampiran:
   a. LKPD-1 sampai dengan LKPD-5
   b. Jurnal kelompok
   c. Evaluasi kelompok

d. Evaluasi pribadi

e. Refleksi pribadi

Kelompok dapat membuat laporan ini menggunakan aplikasi pembuat presentasi atau aplikasi pengolah kata. Selain laporan, pastikan kelompok kalian juga mengumpulkan infografis, peta pikiran, dan poster yang dibuat.

Jurnal kelompok berisi daftar aktivitas individu selama mengerjakan proyek, mulai persiapan s.d perayaan (akhir) proyek yang dapat disusun dengan menggunakan format seperti Tabel 6.11 berikut ini.

Tabel 6.11 Contoh Format Jurnal Kelompok

No.	Nama Aktivitas	Hari/ Tanggal/Jam	Pelaksana	Keterangan
1.	Aktivitas A	Sabtu/13 Februari 2021/10.00	A	Mencari sumber data dari internet
2.				
3.				
4.				
5.				

Ayo Lakukan!

Aktivitas Kelompok

Aktivitas PLB-AD-K11-10: Presentasi Hasil

Proses belajar dalam bab ini dirayakan dengan mempresentasikan hasil belajar kalian yang telah disiapkan dalam bentuk infografis, peta pikiran, maupun poster. Bersama guru kalian dapat menyepakati apakah kegiatan presentasi ini dilakukan dalam kelas kalian saja atau diselenggarakan sebagai pameran di luar kelas sehingga bisa dilihat dan diapresiasi hasilnya oleh teman-teman yang lain atau bahkan oleh masyarakat umum.

B. Pengayaan

Kelompok kalian dapat memperluas proyek ini dengan membandingkan data deforestasi di setiap provinsi dengan data tingkat polusi di provinsi tersebut. Apa yang bisa kalian temukan dari kedua data tersebut jika disandingkan dan dibandingkan?

1. Asesmen dan Refleksi

Kalian dapat melakukan asesmen terhadap hasil kerja kelompok dan masing-masing anggotanya, serta membuat refleksi individu menggunakan panduan pertanyaan berikut. Hasil asesmen dan refleksi selanjutnya disharingkan dalam pertemuan di kelas. Asesmen dan refleksi pribadi ini akan menjadi bahan berharga untuk perbaikan diri kalian masing-masing. Tuliskan secara santun dan konstruktif.

Format Asesmen Proyek

Formulir Asesmen Kelompok Peserta Didik Nama Kelompok : .... Anggota Kelompok : ....
Selama mengerjakan proyek ini, kelompok saya mendapatkan pengalaman: ....
Kendala yang kelompok saya hadapi : ....
Di masa yang akan datang, kelompok kerja saya akan melakukan perbaikan pada: ....

Formulir Asesmen Individu Peserta Didik terhadap Teman Satu Kelompok

No	Nama Peserta Didik	Asesmen kualitatif
1.	A	Peserta didik ini mampu menjadi pemimpin di kelompok ini. Mampu membangun kerjasama dan berkomunikasi dengan baik. Perlu ditingkatkan kemampuan menulisnya. (Ini adalah contoh, kalian dapat menuliskan aspek lainnya yang kalian amati).

No	Nama Peserta Didik	Asesmen kualitatif
2.	B	
3.	C	
4.	D	
5.	E	

Format Refleksi Individual

1. Refleksi individual tentang "Proyekku"

Aspek	Refleksiku
1. Secara keseluruhan, saya tertantang, puas dan senang mengerjakan proyek ini? Mengapa?	
2. Apakah proyekku keren? Bisakah kalian jelaskan bagian terkeren dari proyek kalian?	
3. Pengalaman apa yang paling mengasyikkan dalam proyek ini?	
4. Pengalaman apa yang paling tidak menyenangkan dalam proyek ini?	

Aspek	Refleksiku
5. Apakah saya senang dengan peran saya? Andaikata boleh memilih peran lain, peran apa yang saya inginkan?	
6. Apakah saya puas dengan hasilnya? Adakah yang masih perlu saya perbaiki dan saya sempurnakan?	

2. Berdasarkan pengalaman mengerjakan proyek ini, tuliskan refleksi kalian tentang hubungan proyek yang dikerjakan dengan konsep elemen pengetahuan mata pelajaran Informatika dan bidang lain. Jelaskan dengan ringkas!

Konsep	Yang dibutuhkan untuk menyelesaikan proyek ini, yaitu untuk:
BK	
TIK	
SK	
JKI	
AD	
AP	
DSI	
Bidang lain:.................	

Glosarium

abstraksi (*abstraction*) Merupakan salah satu elemen Berpikir Komputasional yang mengurangi kompleksitas dengan memusatkan perhatian pada gagasan utama dari suatu hal. Dengan menyembunyikan detail yang tidak relevan dengan pertanyaan yang ada dan menyatukan detail yang terkait dan berguna, abstraksi mengurangi kompleksitas dan memungkinkan seseorang untuk fokus pada masalah.

akurasi benar, tidak ada kesalahan.

alamat IP (*IP address*) sebuah identitas angka yang digunakan semua perangkat komputer agar saling berhubungan dalam jaringan internet.

algoritma (*algorithm*) suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan dalam bentuk program komputer untuk menyelesaikan suatu permasalahan komputasi tertentu.

algoritma *greedy* (*greedy algorithm*) setiap algoritma yang berusaha mencapai solusi suatu permasalahan dengan membuat pilihan lokal yang optimal pada setiap tahap.

American Standard Code for Information Interchange (ASCII) tabel standar berukuran tujuh bit yang merepresentasikan karakter huruf non-kapital, huruf kapital, angka, dan karakter lainnya dalam komputer.

analisis data (*data analytics*) proses inspeksi, pembersihan dan pemodelan data dengan tujuan menemukan informasi yang berguna, menginformasikan kesimpulan dan mendukung pengambilan keputusan.

antarmuka grafis pengguna (*graphical user interface*) antarmuka yang menggunakan menu grafis untuk memudahkan pengguna berinteraksi dengan komputer.

antarmuka pengguna (*user interface*) cara pengguna dan sistem komputer berinteraksi, khususnya dengan menggunakan alat input dan perangkat lunak.

antrean (*queue*) struktur data abstrak dalam pemrograman yang merupakan koleksi sekumpulan elemen, di mana penambahan elemen dilakukan di paling ujung yaitu pada "ekor" (tail), dan penghapusan elemen dilakukan pada titik pelayanan yang disebut "head. oleh karena itu antrean juga disebut first in first out (FIFO).

aplikasi desktop (*desktop application*) perangkat lunak yang dibuat untuk dapat dijalankan pada komputer bertipe desktop.

aplikasi mobile (*mobile application*) perangkat lunak yang dibuat untuk dapat dijalankan pada perangkat bergerak.

aplikasi web (*web application*) perangkat lunak yang dapat dijalankan pada suatu server dan dapat dijalankan di menggunakan peramban web.

App Inventor adalah lingkungan pemrograman visual yang intuitif yang memungkinkan semua orang, bahkan anak-anak, untuk membangun aplikasi yang berfungsi penuh untuk smartphone dan tablet Android dan iOS. App Inventor awalnya dikembangkan oleh Google, dan saat ini dikelola oleh Massachusetts Institute of Technology.

asumsi perkiraan, menerima begitu saja.

badan program (*body program/program definition*) realisasi dari fungsi, kode program yang dituliskan di dalam sebuah fungsi.

bahasa pemrograman (*programming language*) kumpulan perintah, instruksi, dan sintaks lain yang digunakan untuk membuat suatu program.

berpikir kritis (*critical thinking*) seni menganalisis dan mengevaluasi pemikiran dengan maksud untuk memperbaikinya.

coding kegiatan menulis kode sumber program.

data (*data*) fakta yang dikumpulkan dan digunakan untuk referensi atau analisis. Data bisa digital atau nondigital dan bisa dalam berbagai bentuk, termasuk angka, teks, uluran tangan, gambar, suara, atau video.

debugging Proses identifikasi dan perbaikan suatu kesalahan pada kode program.

deforestasi peristiwa hilangnya hutan alam beserta dengan atributnya yang diakibatkan oleh penebangan hutan

diagram alir (*flowchart*) sebuah bagan atau diagram dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail serta hubungan antar proses.

dokumentasi (*documentation*) perangkat lunak adalah teks atau ilustrasi tertulis yang menyertai perangkat lunak komputer atau disematkan dalam kode sumber. Dokumentasi menjelaskan bagaimana perangkat lunak beroperasi atau bagaimana menggunakannya, dan mungkin memiliki arti yang berbeda bagi orang-orang dalam peran yang berbeda.

ekspresi (*expression*) komponen bahasa pemrograman yang mendefinisikan suatu proses perhitungan/komputasi.

elemen berpikir komponen-komponen dalam pikiran, yang memungkinkan identifikasi masalah dalam proses berpikir

empati intelektual kemampuan seseorang untuk secara aktif menempatkan diri mereka pada posisi orang lain dalam hal bagaimana mereka berpikir dan merasa.

graf (*graph*) suatu struktur dari sekumpulan objek di mana beberapa pasangan objek memiliki hubungan atau keterkaitan tertentu.

Ilmu Komputer (*Computer Science*) studi tentang komputer dan proses algoritmik, termasuk prinsip-prinsipnya, desain perangkat keras dan perangkat lunaknya, implementasinya, dan dampaknya terhadap masyarakat.

implikasi konsekuensi logis, hubungan antar pernyataan yang bernilai benar ketika salah satu pernyataan secara logis "mengikuti" pernyataan lainnya.

infografis (*infographics*) adalah representasi visual (grafis) dari suatu informasi, data, atau pengetahuan untuk menyajikan informasi yang dapat disajikan dengan cepat dan jelas; biasanya menggunakan elemen grafis untuk menyajikan informasi dengan cara yang menarik secara visual.

informatika (*informatics*) ilmu yang mempelajari penggunaan komputer untuk mengatur dan menganalisis data yang berukuran besar.

inklusif (*inclusive*) dalam konteks Matematika dan Informatika, inklusif berarti 'termasuk'. keterangan 1 sampai 100 (inklusif) artinya kalian dapat memilih bilangan 1, 100, dan semua bilangan di antara 1 dan 100.

***input*/masukan** data yang diterima oleh program untuk diproses.

integritas intelektual sifat, atau keadaan yang menunjukkan kesatuan yang utuh antara pikiran dan tindakan sehingga memiliki potensi dan kemampuan yang memancarkan kewibawaan dan kejujuran

interpretasi kesimpulan, solusi.

inferensi kesimpulan, solusi.

internet (*internet*) jaringan komputer global yang saling berhubungan dengan menggunakan paket protokol internet untuk berkomunikasi dengan jaringan dan perangkat-perangkat yang saling terhubung.

internet of things (IoT) kemampuan terhubungnya benda dan perangkat (misalnya penyiram tanaman, perangkat sensor, dan peralatan sehari-hari lainnya) dengan jaringan yang memungkinkan pengiriman informasi antar-benda menggunakan internet.

jaringan komputer (*computer network*) kumpulan dari dua atau lebih komputer yang dihubungkan bersama-sama untuk tujuan berbagi informasi, dan sumber daya, antara satu sama lain.

jaringan saraf tiruan (*artificial neural network*) sistem adaptif yang dimodelkan berdasarkan jaringan saraf manusia yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut.

kabel *unshielded twisted pair* (UTP cable) salah satu perangkat keras komputer berupa kabel yang digunakan sebagai penghubung atau transmisi data pada sebuah jaringan.

karakter (*character*) dalam pemrograman merupakan suatu tipe data yang menyimpan huruf, angka, spasi, tanda baca atau simbol yang dapat direpresentasikan menggunakan standar kodifikasi tertentu seperti ASCII atau Unicode.

karakter intelektual kebiasaan pikiran, sebagai hasil penerapan standar intektual terhadap elemen-elemen berpikir.

kasus uji (*test case*) suatu kumpulan nilai dengan kondisi tertentu yang dimasukkan ke dalam program oleh penguji

untuk menentukan apakah program yang diuji memenuhi spesifikasi atau berjalan dengan benar.

kebutuhan (*requirement*) mendeskripsikan apa yang akan dilakukan dan bagaimana kinerja yang diharapkan dari suatu perangkat lunak.

kecerdasan artifisial/kecerdasan buatan (*artificial intelligence*) adalah kemampuan komputer untuk melakukan tugas-tugas yang biasanya dilakukan oleh manusia karena memerlukan kecerdasan dan kejelian manusia; kecerdasan yang ditunjukkan oleh mesin yang biasanya dimodelkan dari kecerdasan yang ditunjukkan oleh manusia atau makhluk hidup lainnya.

keluaran (*output*) hasil yang diperoleh dari suatu program yang berjalan yang dikirimkan oleh program, misalnya kepada manusia atau program lainnya.

kepala paket (*packet header*) – bagian dari paket data yang berisi alamat IP pengirim dan penerima, termasuk nomor paket yang memungkinkan penyusunan kembali paket data

klasifikasi proses untuk mengenali, membedakan, dan memahami suatu objek, kemudian mengelompokkannya ke dalam suatu kelas.

knapsack sebuah permasalahan optimasi komputasional dimana kita diberikan sejumlah barang dengan bobot dan nilai masing-masing, serta sebuah wadah dengan kapasitas tertentu, dan kemudian kita ingin mencari skema pengambilan barang ke dalam wadah sedemikian rupa sehingga total nilai barang yang terambil adalah sebesar mungkin dengan total bobot yang masih berada di bawah kapasitas. Permasalahan knapsack memiliki beberapa variasi di antaranya 0-1 knapsack (barang hanya bisa

diambil atau tidak diambil) dan rational knapsack (barang dapat diambil sebagian).

kode sumber (*source code*) bentuk program yang diberikan kepada kompilator untuk dikonversi menjadi object code.

kompleksitas (*complexity*) Jumlah sumber daya minimum, seperti memori, waktu, atau pesan, yang diperlukan untuk menyelesaikan suatu masalah (problem) atau menjalankan suatu algoritma.

kompleks (*complex*) mengandung beberapa unsur yang pelik, rumit, sulit, dan saling berhubungan.

kondisional (*conditional*) sebuah fitur dari bahasa pemrograman yang melakukan perhitungan atau tindakan yang berbeda tergantung pada suatu kondisi benar-salah yang diperiksa.

konsekuensi akibat (dari suatu perbuatan, pendirian, dan sebagainya)

konseptual (*conceptual*) terkait dengan atau berupa suatu ide atau gagasan;

larik (*array*) larik adalah suatu tipe data terstruktur yang dapat menyimpan banyak data dengan suatu nama yang sama dan menempati tempat di memori yang berurutan serta bertipe data sama pula dan dapat diakses berdasarkan indeksnya.

lingkungan pengembangan terintegrasi (*integrated development environment*) program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak seperti editor, kompilator, linker, dan debugger.

logika secara keseluruhan masuk akal, tidak ada kontradiksi,

membaca Kemampuan peserta didik untuk memahami, memaknai, menginterpretasi, dan merefleksi teks sesuai tujuan dan kepentingannya untuk mengembangkan pengetahuan, keterampilan, dan potensinya.

memirsa merupakan kemampuan seseorang untuk memahami, memaknai, menginterpretasi, dan merefleksi sajian visual dan/atau audiovisual sesuai tujuan dan kepentingannya untuk mengembangkan pengetahuan, keterampilan, dan potensinya.

memoisasi (*memoization*) sebuah teknik atau cara untuk menyimpan hasil perhitungan

yang telah diperoleh sebelumnya, agar jika diperlukan lagi, tidak perlu dihitung kembali.

node pada jaringan komputer adalah tahapan dalam jaringan yang dapat menerima dan mengirimkan paket data; router adalah node di jaringan komputer.

pada artificial neural network adalah unit pemroses data yang diinspirasi dari neuron otak manusia.

observasi peninjauan secara cermat, perolehan informasi dari sumber pertama.

otonomi intelektual kemampuan seseorang untuk menggunakan keterampilan berpikir kritis dan mempercayai kemampuannya sendiri dalam berpikir kritis.

peladen (*server*) komputer atau program komputer yang didedikasikan untuk serangkaian tugas tertentu yang menyediakan layanan ke komputer atau program lain di jaringan.

pembelajaran mesin (*machine learning*) Pembelajaran mesin adalah cabang dari kecerdasan buatan dan ilmu komputer yang memiliki fokus pada penggunaan data dan algoritma

untuk meniru cara manusia belajar dan secara bertahap meningkatkan akurasinya.

pembelajaran mendalam (*deep learning*) adalah bagian dari pembelajaran mesin, yang pada dasarnya adalah jaringan saraf tiruan dengan tiga atau lebih lapisan. Jaringan saraf ini mencoba untuk mensimulasikan perilaku otak manusia untuk "belajar" dari sejumlah besar data. Jaringan saraf tiruan dengan satu lapisan dapat membuat perkiraan, namun lapisan-lapisan tambahan pada pembelajaran dalam dapat membantu mengoptimalkan dan meningkatkan akurasi.

pemrogram (*programmer*) orang yang melakukan kegiatan pemrograman.

pemrograman (*programming*) aktivitas yang dilakukan untuk menghasilkan suatu program, termasuk analisis kebutuhan, perancangan, implementasi, dan pengujian suatu program.

pemrograman dinamis (*dynamic programming*) sebuah strategi penyelesaian masalah optimasi komputasional yang bersifat rekursif, dimana solusi permasalahan awal didapatkan dengan menggabungkan solusi dari sub-sub soal permasalahan awal tersebut, namun dengan menghindari adanya redundansi/pengulangan perhitungan dengan memanfaatkan teknik memoisasi.

pencarian (*searching*) suatu jenis permasalahan pada komputasi yang mencari suatu objek yang memenuhi kriteria tertentu dari sekumpulan objek.

penelusuran kode (*code tracing*) menginterpretasikan hasil dari setiap baris kode dan melacak dengan manual efek dari setiap pernyataan.

pengujian (*testing*) pengujian adalah aktivitas untuk memeriksa apakah program yang dihasilkan sesuai dengan persyaratan yang diharapkan dan untuk memastikan

bahwa program bebas dari cacat dan dapat berfungsi sesuai dengan spesifikasi.

pengurutan (*sorting*) suatu jenis permasalahan pada komputasi untuk menyusun kembali suatu himpunan objek secara terurut berdasarkan kriteria tertentu.

penyelesaian masalah (*problem solving*) menggunakan suatu metode teratur untuk menemukan solusi dari suatu permasalahan.

perancangan antarmuka pengguna (*user interface design*) proses untuk merancang antarmuka pengguna dalam perangkat lunak atau perangkat komputasi.

perangkat keras (*hardware*) komponen fisik yang menyusun sistem komputasi, komputer, atau perangkat komputasi.

perangkat lunak (*software*) program yang berjalan di atas sistem komputasi, komputer, atau perangkat komputasi lainnya.

permasalahan (*problem*) sesuatu yang sulit untuk dihadapi atau dipahami.

permasalahan generik/problem generik (*generic problem*) sebuah permasalahan yang bersifat umum dan merupakan hasil abstraksi dari permasalahan-permasalahan yang spesifik, sehingga solusi dari permasalahan generik dapat diterapkan pada berbagai permasalahan spesifik yang masih relevan.

perulangan (*loop*) struktur pemrograman yang mengulangi urutan instruksi selama kondisi tertentu bernilai benar.

prediksi suatu pernyataan tentang data atau kejadian di masa mendatang. Seringkali berdasar pengalaman atau pengetahuan, meskipun tidak selalu demikian.

presisi tepat, rinci hingga level tertentu.

program (*program*) sekumpulan pernyataan yang dapat dieksekusi oleh komputer untuk menghasilkan perilaku yang diinginkan dari komputer.

pseudocode deskripsi program informal yang tidak mengandung sintaks kode atau pertimbangan teknologi yang mendasari

pustaka (*library*) kumpulan kode yang telah ditulis sebelumnya dan dapat digunakan pemrogram untuk membuat program dengan lebih efisien.

rekursi (*recursion*) suatu bentuk pendefinisian sebuah struktur yang mengandung dirinya sendiri. Fungsi/barisan rekursif adalah fungsi/barisan di mana nilainya tersebut ditentukan/tergantung dari nilai fungsi/barisan itu sendiri, pada urutan nilai-nilai sebelumnya.

rekursif (*recursive*) memiliki sifat atau mengandung rekursi.

rekurensi (*recurrence*) relasi yang bersifat rekursif atau mengandung rujukan terhadap dirinya sendiri.

relevansi terkait dengan hal yang dibahas.

router perangkat jaringan yang meneruskan paket data antar jaringan komputer.

signifikansi penting, menjadi konsekuensi

simpul (*node*) unit pokok (elemen) yang merepresentasikan suatu objek yang membentuk suatu graf

standar intelektual standar penting yang diperlukan untuk menumbuhkan kecerdasan dan menjalani kehidupan yang rasional, dapat menjadi alat bantu untuk mengukur kualitas berpikir seseorang

Standard Template Library (STL) adalah pustaka perangkat lunak yang awalnya dirancang oleh Alexander Stepanov untuk bahasa pemrograman C++. STL menyediakan empat

komponen yang disebut algoritma, wadah (container), fungsi, dan iterator.

strategi (*strategy*) langkah terstruktur yang dilakukan untuk melakukan sesuatu; strategi ini dapat ditulis juga sebagai sebuah algoritma.

string Urutan huruf, angka, dan/atau simbol lainnya. Sebuah string dapat mewakili data seperti nama, alamat, atau judul lagu.

struktur kontrol (*control structure*) struktur pada program yang mengimplementasikan suatu kontrol, misalnya kondisional dan perulangan (loop).

teknologi informasi dan komunikasi (*information and communication technology*) istilah umum yang mencakup semua perangkat komunikasi, meliputi radio, televisi, telepon seluler, komputer dan perangkat keras jaringan, sistem satelit dan sebagainya, serta berbagai layanan dan peralatan yang menyertainya seperti konferensi video dan pembelajaran jarak jauh yang memungkinkan pengguna untuk mengakses, menyimpan, mengirimkan, memahami, dan memanipulasi informasi.

tipe data (*data type*) Pengelompokan data yang dibedakan berdasarkan atributnya dan jenis operasi yang dapat dilakukan padanya. Beberapa tipe data yang dasar (atau primitif) adalah integer (bilangan bulat), string, Boolean (benar atau salah), dan floating-point (desimal).

topologi (*topology*) Konfigurasi fisik dan logika suatu jaringan; pengaturan jaringan, termasuk node dan link penghubungnya.

tumpukan (*stack*) struktur data abstrak yang terdiri atas sekumpulan elemen homogen dan didasarkan pada prinsip terakhir masuk pertama keluar (last in first out - LIFO)

ujung paket (*packet trailer*) – bagian dari paket data yang menunjukkan akhir paket data dan cara pemeriksaan kesalahan

Unicode standar teknis yang dirancang untuk mengizinkan teks dan simbol dari semua sistem tulisan di dunia untuk ditampilkan dan dimanipulasi secara konsisten oleh komputer.

vector (STL) salah satu bagian dalam STL bahasa C++ yang dapat menyimpan banyak elemen seperti larik dengan ukuran yang dinamis, menyesuaikan dengan banyaknya data yang tersimpan di dalamnya.

visualisasi representasi grafis dari data, umumnya dipergunakan sebagai cara efisien untuk mengkomunikasikan data dalam jumlah banyak

Daftar Pustaka

__________. (n.d.), Digital literacy, diakses dari en.wikipedia.org/wiki/Digital_literacy pada tanggal 10 November 2021

__________. (n.d.), Critical Thinking, diakses dari westernsydney.edu.au/__data/assets/pdf_file/0006/1082382/Critical_Thinking.pdf pada tanggal 26 November 2021.

__________. (n.d.). What is digital literacy, diakses dari westernsydney.edu.au/studysmart/home/study_skills_guides/digital_literacy/what_is_digital_literacy pada tanggal 10 November 2021

__________.(n.d.). Decision-making process, diakses dari umassd.edu/media/umassdartmouth/fycm/decision_making_process.pdf pada tanggal 29 November 2021

_______. (2020). One Tree Planted. What is Deforestation. Diakses dari youtu.be/vJnnrpSDWPI pada 15 November 2021.

_______. (2017). National Geographic. Climate 101: Deforestation. Diakses dari youtu.be/Ic-J6hcSKa8. Pada 12 November 2021

________. (7 Desember 2020). Angka Deforestasi Netto Indonesia Di Dalam Dan Di Luar Kawasan Hutan Tahun 2013-2019 (Ha/Th). Diakses dari bps.go.id/statictable/2019/11/25/2081/angka-deforestasi-netto-indonesia-di-dalam-dan-di-luar-kawasan-hutan-tahun-2013-2019-ha-th-.html pada 12 November 2021.

__________. (n.d.), Digital literacy, diakses dari en.wikipedia.org/wiki/Digital_literacy pada tanggal 10 November 2021

__________. (n.d.), Critical Thinking, diakses

dari westernsydney.edu.au/__data/assets/pdf_file/0006/1082382/Critical_Thinking.pdf pada tanggal 26 November 2021.

__________. (n.d.). What is digital literacy, diakses dari westernsydney.edu.au/studysmart/home/study_skills_guides/digital_literacy/what_is_digital_literacy pada tanggal 10 November 2021

__________.(n.d.). Decision-making process, diakses dari umassd.edu/media/umassdartmouth/fycm/decision_making_process.pdf pada tanggal 29 November 2021.

__________. (n.d.), Robot Medis, diakses dari id.wikipedia.org/wiki/Robot_medis pada tanggal 5 November 2021

__________. (n.d.). What is a Transmission Control Protocol TCP/IP Model?. Diakses dari fortinet.com/resources/cyberglossary/tcp-ip pada tanggal 5 November 2021.

__________. (n.d). App Inventor Tutorial, diakses dari http://appinventor.mit.edu/explore/sites/all/files/hourofcode/TalkToMePart1.pdf, pada tanggal 10 November 2021

__________. (n.d). Introduction to Machine Learning: Image Classification, diakses dari https://appinventor.mit.edu/explore/resources/ai/image-classification-look-extension, pada tanggal 10 November 2021

__________. (n.d). Voice Calculator Tutorial, diakses dari https://appinventor.mit.edu/explore/resources/ai/voice-calculator, pada tanggal 10 November 2021

__________. (n.d). I have a dream tutorial, diakses dari http://www.appinventor.org/content/ai2apps/simpleApps/dream, pada tanggal 10 November 2021

________. (n.d). Group 5's Teaching Resource Package. http://www.southampton.ac.uk/~mwra1g13/msc/comp6046/Teaching_Resource/resource.html

Agustini, P. (2021 Mei 3), Kementerian Komunikasi dan Informatika, diakses melalui aptika.kominfo.go.id/2021/05/kominfo-catat-1-733-hoaks-covid-19-dan-vaksin pada tanggal 10 November 2021.

Alexander, H.B., (2020), Hebatnya China, Rumah Sakit Corona Dilengkapi Robot Medis, diakses melalui properti.kompas.com/read/2020/02/03/234056221/hebatnya-china-rumah-sakit-corona-dilengkapi-robot-medis pada tanggal 5 November 2021.

Annur, C.M. (2021 Mei 11), Katadata, diakses melalui katadata.co.id/ariayudhistira/analisisdata/609a43a46aa5e/pencurian-data-pribadi-dalam-pusaran-bisnis-fintech-ilegal pada tanggal 10 November 2021.

Baker, Dennis, et.al.,(2001), Guidebook to Decision Making Methods, Department of Energy, United States of America, diakses melalui researchgate.net/publication/255621095_Guidebook_to_Decision-Making_Methods pada tanggal 10 November 2021.

Booth, W. (n.d.) Rainforest Deforestation and Its Effects. Dikases dari youtu.be/Nc7f5563azs pada 15 November 2021.

Carpenter, M., T. Bauer, B. Erdogan, (n.d.), Management Principles v 1.0. , diakses dari 2012books.lardbucket.org/books/management-principles-v1.0/s15-decision-making.html pada tanggal 20 November 2021.

Clarke, John (2019). Critical Dialogues: Thinking Together in Turbulent Times. Bristol: Policy Press. p. 6. ISBN 978-1-4473-5097-2.

Cholle, F.P. (2011 Agustus). What is Intuition and How Do We Use It. psychologytoday.com/us/blog/the-intuitive-compass/201108/what-is-intuition-and-how-do-we-use-it

CPPReference.com. Standard library header <cstring>. en.cppreference.com/w/cpp/header/cstring. Diakses pada tanggal 10 Januari 2021.

CPPReference.com. std::basic_string. en.cppreference.com/w/cpp/header/cstring. Diakses pada tanggal 10 Januari 2021.

CPPReference.com. std::vector. en.cppreference.com/w/cpp/container/vector. Diakses pada tanggal 10 Januari 2021.

CPPReference.com. Array. en.cppreference.com/w/c/language/array. Diakses pada tanggal 10 Januari 2021.

Delima, R., H.B. Santoso, dan J. Purwadi, "Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika", Prosiding Seminar, Seminar Nasional Aplikasi Teknologi Informasi (SNATi), 2016.

Deitel, P. & Deitel, H. (2016). C: How to Program Edisi ke-8

Dhany, F.W.W. (2021 November 24), "Panen Data Pribadi lewat Challenge di Media Sosial", Harian Kompas 24 November 2021.

Edward M. Glaser. "Defining Critical Thinking". The International Center for the Assessment of Higher Order Thinking (ICAT, US)/Critical Thinking Community. Retrieved 22 March 2017.

Forouzan, B. A. (2013). Data Communication and Networking. 5th Ed. New York: McGraw-Hill. ISBN:0073376221

Herdiana, dan Y. Hermawan, "Analisis Dampak Perubahan Revolusi Industri Pertanian 4.0 terhadap Sosial Ekonomi Petani di Kecamatan Praya Barat Kabupaten Lombok Tengah – NTB", Jurnal Media Bina Ilmiah, Vol. 15 no. 4, November 2020, diakses melalui ejurnal.binawakya.or.id/index.php/MBI/article/view/774/pdf

Humas Litbangkes, (2019) Pemanfaatan Teknologi Informasi dalam Kajian Etik, diakses melalui litbang.kemkes.go.id/ pemanfaatan-teknologi-informasi-dalam-kajian-etik/ pada tanggal 5 November 2021

IMD, IMD World Competitiveness Yearbook 2021, diakses melalui imd.org/centers/world-competitiveness-center/ rankings/ pada tanggal 10 November 2021.

Kurnia, T. (2019 Januari 7), Liputan 6, diakses melalui liputan6.com/bisnis/read/3863361/kisah-kerugian-material-akibat-hoaks-di-berbagai-negara pada tanggal 10 November 2021.

Kurniawan,C., (2016), Masa Depan Bidang Kesehatan! Inilah 7 Robot Medis Super Canggih, diakses melalui today.line.me/id/v2/article/Masa+Depan+Bidang+Kesehatan+Inilah+7+Robot+Medis+Super+Canggih-abd8a10c84b8a0b7c9dc4d86c6b2b93e57715ad56c609c73f8c58ef51d084838 pada tanggal 5 November 2021.

Kusnandar, V.B. (2021 Oktober 14), databoks, diakses melalui databoks.katadata.co.id/datapublish/2021/10/14/ pengguna-internet-indonesia-peringkat-ke-3-terbanyak-di-asia pada tanggal 10 November 2021

Miles, B. & Spies-Butcher, B. (2012). Short exercise practice 1: Critical analysis – reading. Sydney: Department of Sociology, Macquarie University.

Oktari, R., (2021), Indonesiabaik, diakses melalui indonesiabaik.id/videografis/indonesia-makin-melek-literasi-digital pada tanggal 10 November 2021.

Paul, R. & L. Elder. (2006). The Miniature Guide to Critical Thinking Concepts and Tools.The Foundation for Critical Thinking.

Subagio, J., (2019), Terinspirasi Transformer, Ahli AS Bikin Robot yang Bisa Lawan Kanker, diakses melalui sains.kompas.com/read/2019/05/27/094638323/terinspirasi-transformer-ahli-as-bikin-robot-yang-bisa-lawan-kanker pada 5 November 2021

Sumartiningtyas, H.N.K., (2020), Robot Medis ini Mengambil Darah Pasien, Akankah Gantikan Peran Dokter?, diakses melalui sains.kompas.com/read/2020/02/10/180300223/robot-medis-ini-mengambil-darah-pasien-akankah-gantikan-peran-dokter- pada 5 November 2021

Tim detikcom (2021 Agustus 7), Detiknews, diakses mealui news.detik.com/berita/d-5673218/terulang-lagi-remaja-tewas-ditabrak-truk-demi-konten-di-tangerang pada tanggal 10 November 2021.

Todd, C. (2015). Deforestation Effects on Climate. Diakses dari youtu.be/AVh2DEgpvsM pada 12 November 2021

Sumber Gambar

Gambar 1.3 Pilar Pengetahuan Informatika. Buku Informatika SMA Kelas VII. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia. Tahun 2020.

Gambar 2.5 Beberapa contoh Rekursi di Kehidupan Sehari-hari. Segitiga Sierpinski commons.wikimedia.org/wiki/File:Sierpinski_triangle.svg

Gambar 2.5 Beberapa contoh Rekursi di Kehidupan Sehari-hari. Boneka bersarang dari Rusia. commons.wikimedia.org/wiki/File:Russian-Matroshka_no_bg.jpg

Gambar 3.4. Proses Pengambilan Keputusan. Sumber: https://2012books.lardbucket.org/books/management-principles-v1.0/s15-decision-making.html

Gambar 3.6 Elemen berpikir. Sumber: dalam Paul dan Elder/2006

Gambar 3.7 Bagan Kaitan Elemen Berpikir, Standar, dan Karakter Intelektual

Sumber: dalam Paul & Elder/2006

Gambar 3.8. Sumber: wartaekonomi.co.id/read286701/rektor-uin-apresiasi-perkembangan-teknologi-pertanian

Gambar 3.9 Buddy the Robot Sumber: 15 Medical Robots That Are Changing the World | IE (interestingengineering.com)

Gambar 5.3 Kecerdasan Artifisial. Sumber: readspeaker.ai/wp-content/uploads/2021/05/deep-learning-vs-machine-learning-readspeaker-840x400.jpg

Gambar 5.4. Ilustrasi Proses Klasifikasi Gambar, Gambar kucing diambil dari

Sumber: By Kari Shea karishea - https://unsplash.com/photos/eMzblc6JmXMImageGallery, CC0, https://commons.wikimedia.org/w/index.php?curid=62177124

Gambar 5.5. Gambar kucing dan anjing dan kelasnya, Sumber: Foto Kucing-1, Oleh Kari Shea karishea - https://unsplash.com/photos/eMzblc6JmXMImageGallery, CC0, https://commons.wikimedia.org/w/index.php?curid=62177124, Foto Kucing 2, https://www.wallpaperhi.com/thumbnails/detail/20200701/5efc68309ab42.jpg, Foto Kucing-3, Oleh Dustin Warrington - Flickr, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=2645977. Foto Anjing-1: https://upload.wikimedia.org/wikipedia/commons/0/0d/A_type_of_dog.jpg, Foto Anjing-2: Oleh Lokal_Profil - Own workshop, CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=811336, Foto Anjing-3: https://upload.wikimedia.org/wikipedia/commons/f/fe/American_Eskimo_Dog_1.jpg

Gambar 5.6. Pengujian dengan gambar baru, Foto Caracal Oleh Derek Keats dari Johannesburg, South Africa - Caracal on the road, early morning in Kgalagadi, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=61699151

Gambar 5.7. Pengujian dengan gambar yang sangat berbeda dari kelas, Foto kuda oleh Larissa Allen - Contact us/Photo submission, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=6642407

Foto Proklamasi Kemerdekaan Republik Indonesia (Aktivitas PLB-AI-K11-02), sumber: By Frans Mendur (also Frans Mendoer) (1913 – 1971) - Photo attributed

to the Department of Information [1]Presidential Documents, National Library of Indonesia [2], Public Domain, https://commons.wikimedia.org/w/index.php?curid=7067766

Foto Ki Hajar Dewantara (Aktivitas PLB-AI-K11-02), sumber https://upload.wikimedia.org/wikipedia/commons/1/16/Ki_Hajar_Dewantara%2C_Kemdikbud.jpg

Foto Bung Tomo ((Aktivitas PLB-AI-K11-02), sumber: https://upload.wikimedia.org/wikipedia/commons/thumb/e/ed/Bung_Tomo.jpg/453px-Bung_Tomo.jpg

Indeks

Auzi Asfarian, S.Komp., M.Kom.

Email : asfarian@apps.ipb.ac.id
Instansi : Institut Pertanian Bogor
Alamat Instansi : Kampus IPB Dramaga, Bogor
Bidang Keahlian : Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Dosen/Peneliti di Departemen Ilmu Komputer IPB (2015 - sekarang).

Riwayat Pendidikan dan Tahun Belajar

- Sarjana Ilmu Komputer, IPB (2012).
- Magister Ilmu Komputer, IPB (2014).

Publikasi

- Usability Evaluation of the Participatory-based KMS Sawit Mobile Application.
- Improving The Usability of Personal Health Record in Mobile Health Application for People with Autoimmune Disease.
- Integrating Humanitarian Technology in Computer Science Education to Internalize Independent Campus Policy: A Case in IPB University.
- Selengkapnya di: scholar.google.com/citations?hl=en&user=Ckl2XTkAAAAJ

Paulina Heruningsih Prima Rosa, S.Si., M.Sc.

Email : rosa@usd.ac.id
Instansi : Universitas Sanata Dharma (USD)
Alamat Instansi : Kampus III, Paingan, Maguwoharjo, Depok Sleman, Yogyakarta 55282
Bidang Keahlian : Informatika / Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Dosen Prodi Informatika USD : 2008 – sekarang.
- Wakil Dekan I FST USD : 2015 - 2019

Riwayat Pendidikan dan Tahun Belajar

- 1988 -1993: S1 Prodi Ilmu Komputer - Universitas Gadjah Mada
- 1996 -1999: S2 Department of Computer Science - Ateneo de Manila University, Philippines

Publikasi

- Kontributor artikel dalam Buku Kumpulan Hasil Penelitian Tentang Pemilu, Penerbit Universitas Sanata Dharma, 2015.
- P.H.P Rosa, H. Sriwindono, R.A. Nugroho, K. Pinaryanto, 2020, Comparison of Crossover and Mutation Operators to Solve Teachers Placement Problem by Using Genetic Algorithm, Journal of Physics: Conference Series, Vol. 1566, July 2020
- Selengkapnya di: scholar.google.com/citations?user=hq3RtHUAAAAJ

Irya Wisnubhadra

Email : irya.wisnubhadra@uajy.ac.id
Instansi : Universitas Atma Jaya Yogyakarta
Alamat Instansi: Jl. Babarsari 44, Yogyakarta
Bidang Keahlian : Pemrograman, Database System, Business Intelligence

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Dosen Pengajar Tetap, Teknik Informatika Universitas Atma Jaya Yogyakarta

Riwayat Pendidikan dan Tahun Belajar

- S1: Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada.
- S2: Teknik Informatika, Rekayasa Perangkat Lunak, Institut Teknologi Bandung.
- S3: Faculty of ICT, Universiti Teknikal Malaysia, Melaka (2018-sekarang).

Publikasi

- Copyright Perangkat Lunak, Aplikasi monitoring transportasi buah sawit, logtransawit. online, 2019.
- Pengembangan mobility business intelligence untuk peningkatan produktivitas sistem transportasi TBS kelapa sawit secara berkelanjutan, Penelitian Terapan, Tahun 2020 – 2021, DIKTI
- Selengkapnya di: scholar.google.com/citations?hl=en&user=meXn604AAAAJ

Dr. Mushthofa, S.Kom, M.Sc.

Email : mush@apps.ipb.ac.id
Instansi : Institut Pertanian Bogor
Alamat Instansi : Kampus IPB Dramaga, Bogor
Bidang Keahlian : Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Dosen/Peneliti di Departemen Ilmu Komputer IPB (2009 - sekarang)
- Research Assistant di Ghent University, Belgia (2014 - 2018)

Riwayat Pendidikan dan Tahun Belajar

- Sarjana Ilmu Komputer, IPB (2005)
- Master's in Computational Logic, TU Vienna, Austria (2009)
- Doctor in Computer Science/Bioinformatics, Ghent University, Belgia (2018)

Publikasi

- Fuzzy Answer Set Programming: From Theory to Practice (Chapter pada Beyond Traditional Probabilistic Data Processing Techniques: Interval, Fuzzy etc. Methods and Their Applications, 2020).
- Modeling multi-valued biological interaction networks using fuzzy answer set programming, Fuzzy Sets and Systems, Volume 345, 2018.
- Selengkapnya di: scholar.google.com/citations?user=2GOnkGMAAAAJ

Dean Apriana Ramadhan, S.Komp., M.Kom.

Email : deanaprianaramadhan@apps.ipb.ac.id
Instansi : Departemen Ilmu Komputer, FMIPA IPB
Alamat Instansi : Kampus IPB Dramaga, Bogor
Bidang Keahlian : Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Dosen Departemen Ilmu Komputer IPB.
- Asisten Direktur, Direktorat Sistem Informasi dan Transformasi Digital, IPB University,

Riwayat Pendidikan dan Tahun Belajar

- Sarjana Ilmu Komputer, Departemen Ilmu Komputer IPB (2012).
- Magister Ilmu Komputer, Departemen Ilmu Komputer IPB (2015).

Publikasi

- Ramadhan DA, Nurhadryani Y. Hermadi I. 2014. Campaign 2.0: Analysis Of Social Media Utilization In 2014 Jakarta Legislative Election. International Conference on Advanced Computer Science and Information Systems (JCACSIS} 2014. Jakarta (ID): Fasilkom UI.
- Selengkapnya di: scholar.google.com/citations?hl=en&user=Mgk9EuMAAAAJ

Dr. Inggriani

Email : inge@informatika.org
Instansi : Bebras Indonesia, ITB, IT Del
Bidang Keahlian : Informatika

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Anggota Asesor BAN PT (2014-sekarang)
- Anggota Senat Akademik Institut Teknologi Del (2014-sekarang)
- Dosen STEI ITB (1977-2018) – purnabakti

Riwayat Pendidikan dan Tahun Belajar

- 1977: Bachelor of Engineering Physics.
- 1985: Master DESS-IDC (Diplôme d'Etudes Supérieures Spécialisées, Informatique Double Compétence), Université Grenoble I, France.
- 1986: Master DEA Informatique, Institut Nationale Polytechnique de Grenoble, France.
- 1989: Docteur en Informatique, Université Joseph Fourier, Grenoble, France.

Publikasi

- Rouvrais S., Chelin N., Gerwel P. C., Audunsson H., Liem Inggriani., Tudela V. L., "Preparing 5.0 Engineering Students for an Unpredictable Post-COVID World", World Engineering Education Forum and the Global Engineering Deans Council (WEEF/GEDC) Virtual Conference, 16 – 19 November 2020.
- Selengkapnya di: scholar.google.com/citations?hl=en&user=SrCED_oAAAAJ.

Ir. Julio Adisantoso, M.Kom.

Email : julio@apps.ipb.ac.id

Instansi : Institut Pertanian Bogor

Bidang Keahlian : Informatika

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Direktur Sistem Informasi dan Transformasi Digital, Institut Pertanian Bogor
- Dosen Departemen Ilmu Komputer, Institut Pertanian Bogor.

Riwayat Pendidikan dan Tahun Belajar

- 1984 – Bachelor of Science in Statistics. Bogor Agricultural University, Bogor, Indonesia
- 1996 – Master of Computer Sciences. University of Indonesia, Jakarta, Indonesia

Publikasi

- Adisantoso, J., 2021. Pembelajaran di Era Digital: Kesiapan Teknologi Informasi Perguruan Tinggi. Prosiding Transformasi Pembelajaran Nasional Vol 1: Peluang dan Tantangan Pembelajaran Digital di Era Industri 4.0 Menuju Era 5.0. P.1.
- Afriza A, Adisantoso J. Metode Klasifikasi Rocchio untuk Analisis Hoax. Jurnal Ilmu Komputer dan Agri-Informatika. 2018 Jul 25;5(1):1-0.
- Selengkapnya di: scholar.google.com/citations?user=4QSq4oEAAAAJ

Riksa Arif Fitriyani, S.Sn.

Email : riksarifoi@gmail.com
Alamat Kantor : Jl. Pontira 1, Cimahi, Bandung.
Bidang Keahlian : Animator, Desainer, Illustrator.

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- 2020 - 2021 , Web Asset Design (buku.kemdikbud.go.id)
- 2020 , Web Asset Design (ykep.org)

Riwayat Pendidikan dan Tahun Belajar

- 2012 - 2016 , Universitas Bina Nusantara

Lucki Hersya Rachman, S.Pd.

Email : lucki.hersya@student.upi.edu
Instansi : SMPN 3 Margahayu
Alamat Instansi : Jl. Sadang no. 184, Margahayu Tengah, Margahayu, Kab. Bandung
Bidang Keahlian : Pendidikan Ilmu Komputer

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- Guru BTIK, SMPN 3 Margahayu (2019 - sekarang)

Riwayat Pendidikan dan Tahun Belajar

- 2010 - 2014 : S1 Pendidikan Ilmu Komputer, Universitas Pendidikan Indonesia

Judul Penelitian dan Tahun Terbit (10 Tahun Terakhir):

- 2014 : Pengembangan Multimedia Pembelajaran Interaktif Berbasis Metode Problem Based Learning (PBL) pada Materi Entity Relationship Diagram (ERD)

Futri F. Wijayanti, S.Hum., M.A.

Email : futri.fuji@kemdikbud.go.id

Instansi : Pusat Perbukuan, BSKAP, Kemdikbudristek

Alamat Instansi : Jalan RS. Fatmawati Gedung D Kompleks Kemdikbudristek Cipete, Jakarta 12410

Bidang Keahlian : Pengembang Perbukuan

Riwayat Pekerjaan/Profesi (3 Tahun Terakhir):

- 2007-2008, Library Consultant (independen)
- 2008-2010, Legal Librarian (Pamungkas & Partners)
- 2010-sekarang, Pengembang Perbukuan, Penyunting (Kementerian Pendidikan dan Kebudayaan)

Riwayat Pendidikan dan Tahun Belajar

- 2003-2007, Universitas Indonesia
- 2015-2017, Universitas Gadjah Mada

Judul buku dan tahun terbit:

- Biji Semangka Ajaib (2020)

Judul buku yang pernah disunting:

- Rusaknya Suara Kodok (2019)
- Operasi Sampah di Taman (2019)
- Rambut Juga Butuh Mandi (2019)
- Titi dan Ira Berbagi Kebahagiaan (2019)
- Buku Panduan Guru Pendidikan Khusus bagi Peserta Didik dengan Hambatan Intelektual Jenjang SDLB, SMPLB, SMALB (2021)